

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 17/30	A2	(11) International Publication Number: **WO 00/51021** (43) International Publication Date: 31 August 2000 (31.08.00)

(21) International Application Number: PCT/IB00/00314

(22) International Filing Date: 11 February 2000 (11.02.00)

(30) Priority Data:
09/248,811 11 February 1999 (11.02.99) US

(71) Applicant *(for all designated States except US)*: SIGNIFILE B.V. [NL/NL]; Maliesingel 33, NL-3218 BJ Utrecht (NL).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: MAAS, Pascal [NL/NL]; Nobelstraat 28bis, NL-3512 EP Utrecht (NL). VAN DER HART, Jan, Jaap [NL/NL]; M.P. Lindostraat 21, NL-3532 XE Utrecht (NL). BOUCHER, Ernest, Hugues [NL/NL]; Maliesingel 33, NL-3581 BJ Utrecht (NL). BOUCHER, Willem, Lodewijk [NL/NL]; Farkabaan 705, NL-3524 ZG Utrecht (NL).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: ELECTRONIC DOCUMENT STORAGE AND RETRIEVAL SYSTEM AND METHOD

(57) Abstract

A software application and computer interface for indexing and retrieving computer documents. Users index their documents with a principally unlimited number of attribute buttons, each describing the content, context, environment or any other aspect or interesting feature of that document. Indexes are associated by selecting the desired object icons and clicking one or more link icons, situated on the left portion of each attribute button. Once an index is associated, the user may click on the right portion of the attribute button, referred to as an attribute icon, at any time and all files or object icons associated with that attribute will be displayed in the retrieved tab.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

Electronic Document Storage and Retrieval System and Method

Field of the Invention

This invention relates to a software application and computer interface for indexing and retrieving data files.

Background of the Invention

Information technology has drastically improved our ability to manage electronic data in an efficient manner. As such data represents a unique and invaluable form of human thought and intellectual product. It is indeed imperative that tools for storage, maintenance, and retrieval be developed to the fullest extent possible. Yet, despite the sophistication of current data management systems such as those for managing documents, there is still a noticeable absence both in their methodology for manipulating data and ability to satisfactorily interface with the user.

This deficiency in the technological field of data management, specifically, document management can be divided into three essential categories: the hierarchy problem, the publication problem, and the needle in a haystack problem.

When storing a document, computer users typically look for a digital location that best indicates the content or context of that particular document. For

example, documents may be stored by client number, subject, geographic location, date, or author, in order to establish a mental link between a specific document and its place of storage. Thus, computer storage is usually divided into multiple directories, folders, or drawers, each having a specific name or attribute, whereby all documents stored in any one location potentially share a common association such as the name of the folder or drawer. Retrieving a document is therefore a much simpler task since the associations enable users to quickly find the storage location of a particular document.

According to current technologies, however, individual documents are only situated in one place of storage, such as a folder or drawer, and therefore are limited to only one association. Since most documents have many different aspects, the person who stores a document will often have to choose which aspect is most important. Thereafter, retrieving a document is only possible through that chosen aspect or association and if a user recalls a different aspect or association it is to no avail since the document has not been indexed in that manner. For example, if a user stores a business document relating to a licensing agreement by the client number he or she would have a difficult time retrieving all documents relating to licensing agreements, but it would be simple to retrieve all documents relating to that client number.

Currently, most digital documents are stored according to a hierarchical file system. This system is essentially a multi-leveled storage system, wherein any folder or drawer can, in addition, have multiple sub-folders or sub-drawers beneath them. Thus, rather than being limited by one name or attribute as described above, a hierarchical system enables users to associate files with a hierarchical set of drawers. For example, a user can create a drawer titled

'license agreements,' two sub-drawers titled 'local clients' and 'International clients,' and several additional sub-drawers, beneath each of the aforementioned sub-drawers, titled according to the appropriate client number. A document stored in a sub-drawer can now be associated with the name of that drawer and also with the name of any and all parent drawers. In this case, a document may be associated as a licensing agreement for an international client, having a particular client number. Without the hierarchical system this flexibility would not be possible.

When hundreds or thousands of documents accumulate in hierarchical storage, two problems can arise. A large number of drawers and sub-drawers are needed to store all documents. This makes it inefficient and time-consuming to find a document because it requires traversing many drawers one at a time. Secondly, a large number of documents can reside in the same drawer. To find a document, it is required to inspect each individual document to establish if it is the one that was sought.

With a large number of drawers and the limitation that each document can only have one storage location, it may be difficult to choose the appropriate drawer or sub-drawer. Indeed, the same problem as mentioned above occurs here as well. Specifically, each document may have multiple associations or aspects and cannot be stored in only one location. For example, a law firm may desire to store documents by client number, by date, and by area of law. Even hierarchical storage cannot accomplish such a task.

An alternative method to create multiple associations would be to store more than one copy of a document, one per selected drawer or folder. Creating multiple copies of a document, however, causes many problems. If multiple

documents exist it would be difficult to keep data accurate and uniform when one or more copies are modified. Further storage resources are extensively used and the process of creating multiple documents is time consuming and inefficient.

A second deficiency with current document management systems is the publication problem. On a network system files are often shared by many users. The process of sharing documents interchangeably is referred to as publication. The problem here is that each individual on a network may have their own method of storing documents, and other users stay uninformed about the existence and content of other's documents. Since documents can only be stored in one manner it is difficult to establish the relevance of a document without opening and examining it. Furthermore, since each user has his own storage preferences, there is no common structure of reference that enables various users to efficiently share documents.

Similarly, the third problem, the needle in a haystack, needs to be circumvented as well. A system that has many files and many drawers becomes a handicap since unless a user remembers the content and position of each individual file it may be difficult or impossible to subsequently find it. Finding one file in one sub-drawer among hundreds is comparable to finding a needle in a haystack. Several applications have been developed to overcome these problems, but none offer a satisfactory solution. For example, Microsoft® incorporates a Find utility into Windows®. This utility searches all files or documents on a computer storage device for a particular file name or selected text within the searched file. This process is both very slow and inaccurate. Scanning an entire storage device takes time and the Find utility has no method for finding

associations. Further, analyzing text within documents is only possible for text documents and the actual text is often not related to any association or aspect of the document.

Another utility, used frequently on the Internet, is an index server or search engine. Products like AltaVista® or Microsoft Index Server keep a database of all documents and their locations. Simply, these programs perform similarly to Microsoft Find, but they are much faster and more customizable. Nevertheless, these utilities as well are limited to retrieving via the name of a document or text within a document.

Thus, there is a need for a document management system that resolves the limitations of the aforementioned prior art. The required system should provide a common context that can be applied by all users to store, share, and retrieve documents. And other electronic files, referred to as objects.

Object and Summary of the Invention

It is thus a general object of the present invention to provide a software system and computer interface to efficiently index and retrieve objects.

A more specific object of the present invention is to provide a software system and computer interface that enables users to associate objects with multiple attributes and subsequently retrieve them efficiently and without significant trouble, so that the limitations of prior art document management systems, such as the traditional hierarchical file structure, can be avoided.

The system of this invention gives users greater visibility of files in a network or other environment, a means to communicate through files with others, and a

way to control documents in a multi-user business environment. Essentially, users are able to index their documents with a principally unlimited number of attributes, describing the content, context, environment or any other aspect or interesting feature of that document. Thus, rather than using only one identifying feature when referencing an object, as with the prior art, this system uses multiple attributes so that any object can be retrieved via many ways.

The software system according to one embodiment of this invention comprises an interface, which primarily includes an object view and an attribute view. Generally, the object view is a file view, which displays multiple files or directories located on a workstation or network, referred to herein as object icons. The attribute view displays link icons, attribute icons and folder icons, among other items or tools, for organizing and sorting the files or data (represented as object icons) from the object view. The process of organizing these files is achieved by associating them with one or more customized attributes, as will be explained in more detail below.

With reference to the object view, the display of files or documents is based on the selection of one of three tabs: a browse tab, an unindexed tab, and a retrieved tab. The browse tab allows a user to manipulate objects and files similar to the manner objects are manipulated in the environment their invention operates under. For example, in a Microsoft Windows environment the browse tab operates similar to a Microsoft Windows Explorer interface, whereby all files on a workstation or server are visible and the Windows directory structure can be traversed. The unindexed tab is customizable, and typically displays recently created files from certain directories that have not been indexed by the user. This tab operates similar to an in-box, where the user

can see which objects have not been integrated into the association system of this invention. The retrieved tab operates in conjunction with the attribute view, which will be explained below, by displaying all documents associated with a selected attribute icon.

In order to associate an object, such as a file or other data, with an attribute, the user would typically select either the browse or unindexed tab and highlight one or more object icons that he or she wishes to index. Thereafter, the user creates an association between these selected object icons and one or more attribute icons by selecting a link icon, corresponding to each of the attribute icons. Thus, whenever an object icon is highlighted in the object view and the user clicks one or more link icons on the attribute view, an association is created corresponding to the attribute icon associated with each selected link icon. Should a user wish to remove this association between an attribute icon and a document, he or she simply re-selects that document and clicks the link icon.

Once an index is associated, the user may click on the attribute icon at any time and all object icons associated with that attribute icon will be displayed in the retrieved tab. Notably, the user may select multiple attribute icons simultaneously and any object icon that is associated with all selected attribute icons will be displayed. The wording on the attribute icon changes color to show that is has been selected.

The attribute view also contains folder icons, in addition to attribute icons so that the user may sort their attributes according to a customized hierarchical structure as well. These folder icons may be selected and deselected to respectively display or hide all attribute icons or sub-folder icons associated

with that folder icon. Folder icons with a down-facing arrow are selected folder icons, whereby their attribute icons are visible, while the folders with side-facing arrows are not-selected and hence do not display any sub-folder icons or attribute icons. Other features include, sorting attributes and folders alphabetically and displaying only selected attribute icons.

Furthermore, the user is also able to adjust the display criteria of the retrieved tab via tighten and loosen icons on a button bar on the attribute view. For example, the user may simultaneously select attribute icons 'XYZ Consulting,' 'BDO CampsObers,' and 'ABC Bank.' Initially, the retrieved tab will only display object icons associated with all three of those attributes (i.e. files that have been assigned to all three selected attribute icons). The user can, however, adjust the criteria so that all object icons matching at least 2 out of 3 of these attributes or 1 out of 3 of these attributes will be shown, and so on.

The above description sets forth rather broadly the more important features of the present invention in order that the detailed description thereof that follows may be understood, and in order that the present contributions to the art may be better appreciated. Other objects and features of the present invention will become apparent from the following detailed description considered in conjunction with the accompanying drawings. It is to be understood, however, that the drawings are designed solely for the purposes of illustration and not as a definition of the limits of the invention, for which reference should be made to the appended claims.

Detailed Description of the Drawings

In the drawings in which like reference characters denote similar elements throughout the several views:

FIGS. 1a-1c illustrate sample views of the computer interface in accordance to exemplary embodiments of this invention;

FIG. 2 is a sample view of the computer interface of one embodiment of this invention, displaying the browse tab;

FIG. 3 is a sample view of the computer interface of one embodiment of this invention, displaying the unindexed tab;

FIG. 4 is a flow diagram which illustrates the sequence of operation, according to one embodiment of this invention, for creating an index and subsequently retrieving documents under that index;

FIG. 5 illustrates a selected document according to one embodiment of this invention;

FIG. 6 illustrates a selected link icon according to one embodiment of this invention;

FIG. 7 illustrates a link icon, which is turned on, according to one embodiment of this invention;

FIG. 8 illustrates a selected and non-selected attribute icon according to one embodiment of this invention;

FIGS. 9 and 10 show how more than one attribute can be associated to each document or data file;

FIG. 11 illustrates the inactive phase or cue of the link icon according to one embodiment of this invention;

FIG. 12 illustrates the process of making an association between an object and an attribute, according to one embodiment of the invention;

FIG. 13 illustrates how a user can retrieve objects via attribute icons, according to one embodiment of the invention;

FIG. 14 shows how a user can inspect the attribute associations of an object, according to one embodiment of the invention;

FIG. 15 shows how a user can remove an association between an attribute and an object, according to one embodiment of the invention;

FIG. 16 shows how a user can inspect the attribute associations of an object, where the object is associated with a plurality of attributes, according to one embodiment of the invention;

FIG. 17 illustrates the selection of a plurality of objects, according to one embodiment of the invention;

FIG. 18 illustrates the use of a filter icon, according to one embodiment of the invention, to facilitate negation searches or other appropriate manipulation;

FIG. 19 illustrates an example of a prior art file system;

FIG. 20 illustrates the structure of the polymorphous object allocation table (POAT) file system according to one embodiment of the present invention;

FIG. 21 illustrates one embodiment of the invention where the POAT system is structured to control the access of individual users, such as in network environments; and

FIG. 22 illustrates several graphic examples of alternative embodiments of this invention.

Detailed Description of the Invention

FIG. 1a illustrates a user interface 10 in accordance with one embodiment of the present invention. Interface 10, may be displayed on a computer terminal monitor or any other commercially available display device. Object icons 12 provide a visual representation of objects referenced by the

system. It is noted that in case of objects stored in a memory system, the memory system includes memory devices that are disposed locally or remotely from a user, as for example in a computer network system, and as such, the invention is not limited in scope in that respect.

The objects represented by object icons 12 comprise a variety of objects in a variety of media. For example, object icons 12 in accordance with some embodiments of the invention represent: (1) electronic documents; (2) database records; (3) data corresponding to images; (4) references to documents, for example Internet URL addresses; (5) E-mail documents; (6) physical objects, e.g. books; and (7) any other type of object that is stored for later retrieval by users of the system.

In accordance with one embodiment of the invention, user interface 10 is configured as a tool to index and retrieve the objects represented by object icons 12. To this end, attribute icons 14, as illustrated in FIG.1a, provide a visual representation of certain attributes that a user desires to associate with the stored objects as represented by object icons 12. Thus, a user may desire to associate a particular attribute to a plurality of objects. Vice versa, a user may desire to associate a particular object to a plurality of attributes. Advantageously, the attributes provide indications that are meaningful to the user for purposes of associating to objects. For example, these attributes comprise descriptive indices that describe documents stored in a computer system based upon agreed upon terminology as explained in more detail below. Other types of attributes include images or sounds that a user desires to associate with objects. For example an indicative image may be associated with an object relating to an animation. Still other types of attributes comprise

execution commands that a user desires to associate with one or more objects. For example, an attribute icon 14 advantageously represents an execution command that when selected by a user executes predefined codes on executable objects that have been previously associated with that attribute.

In accordance with one embodiment of the invention, in order to provide the user with a convenient means for associating the attributes represented by attribute icons 14 to objects represented by object icons 12, link icons 16 are provided that are visually disposed next to or adjacent to attribute icons 14, although the invention is not limited in scope in that respect. For example link icons 16, in accordance with another embodiment of the invention are merged with attribute icons 14 as will be explained in more detail in reference with FIG. 1b.

Link icons 16 provide a visual indication to a user who desires to set a link between an object represented by object icon 12 and an attribute represented by attribute icon 14. A link may be set, for example, by a cursor pointer in conjunction with a computer mouse. The user may move the mouse so that the cursor pointer moves over a desired object icon, such as 12c. The user may then click on the mouse once to highlight the desired object icon. Thereafter the user may drag the mouse so that the cursor pointer moves over a desired link icon, for example 16a. By clicking the mouse, the system sets a link between a desired object represented by object icon 12c and an attribute represented by attribute icon 14a corresponding to its adjacent selected link icon 16a.

The user may repeat the tasks described above, consecutively, to set another link between object icon 12c and, for example, attribute icon 14c. Every time a

link icon is selected by the user in accordance with the tasks described above, the system provides a visual indication of the corresponding association that is set between the object icon and attribute icon. One form of this visual indication may include a change of color of selected link icons as illustrated in FIG. 1a. Clicking on a link icon that already indicates an association removes the association between the selected object(s) and the attribute.

Illustratively, in accordance with one embodiment of the invention, the task of making an association between an object represented by object icon 12a and an attribute represented by attribute icon 14c, is shown in fig. 12. To do this, both object icon 12a and attribute icon 14c must be visible, as can be seen in step 11 of figure 12. Then, desired object icon 12a is selected as shown in step 13. Thereafter, as is shown in step 15, desired link icon 16c can be selected to set the association between the object, represented by object icon 12a, and the attribute, represented by attribute icon 14a.

Once the aforementioned associations are created the user can then accomplish desired tasks on objects identified by object icons 12 during a control and/or retrieval mode. During this control or retrieval mode, the user selects a desired attribute, for example, by moving a cursor pointer over one or more desired attribute icons and highlights the selected attribute icon, for example by clicking the mouse. The system then shows all object icons that have been previously associated with the selected attribute icon(s). In one embodiment of the invention where a selected attribute icon represents a code-execution command, the system executes the executable codes represented by the attribute icon on objects represented by object icons associated with the selected attribute icon(s).

Fig 13 illustrates further how a user can retrieve objects by attributes, as specified above. As illustrated at step 19, the user selects a desired attribute 14c. The system subsequently searches for all objects associated with the selected attribute and displays them as objects, as shown in Step 21, where in this example object icon 12a is displayed.

With reference to figure 14, in accordance with one embodiment of the invention, the user can inspect the attribute associations of an object. Inspecting the attribute associations of an object enables users to determine which attributes are associated with that object. To do this, both the object icon representing an object and the attribute icon representing an attribute must be visible, as can be seen in step 23 of figure 14, wherein object icon 12a and attribute icon 14c are illustrated. The user then selects objects he wishes to inspect, such as object icon 12a at step 25. Thereafter the system shows the associated attribute icons by automatically displaying or highlighting the appropriate link icons, such as link icon 16c, as shown in step 27.

Figure 16 shows the same operation, this time when an object is associated to a plurality of attributes. Thus, at step 37 the user begins to inspect the attributes that have already been associated with a desired object, such as an object represented by object icon 12a. At step 39, the user selects object icon 12a, in response to which the system highlights it to the user. Thereafter, the system, at step 41, indicates all the associated attribute icons by automatically displaying and highlighting the appropriate link icons, such as link icons 16c, 16e and 16f.

In reference to figure 15, at step 29, the user can begin to remove an association between an attribute and an object in the following manner. At step 31, the

user selects an object represented by an object icon, such as object icon 12a. At step 33, the system shows the associated attribute icons by displaying the appropriate link icons, such as link icon 16c. At step 35, the user deselects link icon 16c, causing the association to be removed, by clicking over icon 16c. Figure 17 shows what happens when a user selects more than one object to simultaneously associate multiple selected objects with one or more common attributes. At step 43, the user selects one object, shown here as object icon 12a, and the system responds by displaying and highlighting the appropriate link icons, such as link icons 16c, 16e and 16f. Similarly, at step 45, the user selects another object, shown here as object icon 12b, and the system responds by displaying and highlighting the appropriate link icons, such as link icons 16d and 16g. At step 47, the user selects both object icons 12a and 12b, whereby the system responds by displaying and highlighting *in shaded form* the appropriate link icons, such as link icons 16a-16g. The system highlights a link icon in shaded form when its attribute icon is associated with one or more of the selected objects but not with all of them. Neither a highlighted or non-highlighted link icon would be appropriate in this situation since neither of these formats correspond to the status of all selected object icons. Since different objects do not need to carry the same associations, it is not always possible to display the associations of a plurality of objects at the same time. Thus, by any multiple selection of objects, associations between objects and attributes can be made, but not necessarily inspected.

At step 49, however, the user selects link icon 16d, causing an association to be created with attribute icon 16d and all selected object icons, namely object icons 12a and 12b. Notably, since an association already exists with object

icon 12b, no adjustment is necessary for that object. Now, since all selected objects are associated with attribute icon 16d, the corresponding link icon is shown highlighted and not in shaded form. Similarly, the user may deselect link icon 16d and remove the association between attribute icon 16d and object icons 12a and 12b. This is shown at step 51, whereby link icon 16d is displayed in a non-highlighted format, corresponding to selected object icons 12a and 12b both not having an association with object icon 16d.

In accordance with another embodiment of the invention that enables negation in searches, the user can manipulate the way in which objects can be retrieved. In addition to being able to specify attributes for the retrieval of objects, the user can also choose for certain associations not to be shown. Figure 18 shows an attribute view 1800 and object view 1802. In addition to attribute and link icons, such as attribute icons 14a and 14b and link icons 16a and 16b, the attribute view further comprises supplemental icons for performing boolean and other related searches, such as filter icons 1804a and 1804b. Filter icons are used here to manipulate the filtering of objects when doing a retrieve action.

At step 53, the associations for selected object 12a are shown. At step 55, the association for selected object 12b is shown. At step 57, attribute icon 14a is selected, which causes the system to show associated objects 12a and 12b in the object view. At step 59, the user selects a filter icon, such as filter icon 1804b, which, at step 61, immediately causes its corresponding attribute icon to become selected as well. As a result of selecting filter icon 1804b object icon 12b is shown on the object view. This result occurs because the filter icon acts as a negation toggle. Thus, the object view displays all

object icons associated with selected attribute icon 14a and NOT associated with 'filter selected' attribute icon 14b. During a retrieval process, if the filter toggle, such as filter icons 1804a and 1804b, is selected, objects associated with the corresponding attribute are excluded from display in the object view. If the toggle is not selected, objects associated with the corresponding attributes are selected for display.

FIG. 1b illustrates a user interface 20 in accordance with another embodiment of the invention. In this embodiment link icons 26 are disposed within the area defined by attribute icons 14. For example, link icons 26 may be embedded as a visual characteristic within various characters or letters associated with attribute icons 14. This visual characteristic may be the shading or coloring of the characters or letters. The visual characteristics may also be a change in the visual representation of images or letters displayed within the area defined by attribute icon 14.

For the embodiment illustrated in FIG. 1b, the user associates an object icon with an attribute icon by first highlighting a desired object icon as described above in reference with FIG. 1a and then, moving, for example, the cursor pointer over an attribute icon and right click on the mouse. In response, the system changes the visual characteristic of letters or images within the attribute icon so as to provide an indication that the attribute represented by attribute icon 14 is associated with an object represented by selected object icon 12. For example, the color of letters within an attribute icon change from one color to the other to indicate that an association with a selected object icon has been made. Thus, the letters or images within, or associated with an attribute icon function both as a description of the attributes that are being associated with an

object icon and also as a link icon that can be highlighted or modified to indicate an association with an object icon.

The system of this invention, as initially described in the aforementioned embodiments, can be implemented as a complete file system for more efficient storage and retrieval of data on a storage device or other similar structure. Normally, with file systems, such as those under DOS or Windows, objects (e.g. files in a file-system) are represented in a linear fashion. Each object has a unique identifier (e.g. a filename) and a location reference to that object (e.g. a track and sector number), indicating where the object is stored. In addition, attributes may be added like the creation date, object size or a security descriptor.

Figure 19 illustrates an exemplary prior art system with regard to object control. Column 1901 depicts the actual objects, column 1903 contains the object identifiers, column 1905 contains the object location references, and column 1906 contains additional object attributes. A unique object identifier (from 1903) is linked with a unique object location reference (from 1905) and optionally with an object attribute (from 1906). Any additional object attribute would require an additional column. The table with columns 1903, 1905 and 1906 is further referred to as the Object Allocation Table (OAT). Thus, each individual object is addressed from the Object allocation table (OAT) with *one* unique object identifier.

Addressing objects stored in a logical memory is done in either of two ways; in case of fixed length ordered data objects the location or the start address or 'offset,' represented by small black squares 1904, is calculated based on the position of the object identifier in the Object allocation table (OAT). Hence,

the location reference column 1905 is sometimes referred to as a virtual (calculated) column. In case of objects of differing length the position of the object's offset cannot be determined by calculation and, therefore, is saved as an attribute in a dedicated attribute column in the Object allocation table (OAT). Handling large numbers of objects creates a need for faster access and more detailed description and user identification of objects. To meet this demand an Object allocation table (OAT) can be used to address another Object allocation table (OAT) that addresses objects in its turn. This is referred to as a hierarchical Object allocation table (HOAT). The amount of HOAT's underlying a principal Object allocation table (OAT) is theoretically unlimited. For convenience we the term OAT refers to both OAT and HOAT. Practical examples of Object allocation table (OAT)'s at the file system level are FAT (File Allocation Table), NTFS (New Technology File System), and FAT32 (32-bit File Allocation Table), which are used, for example, with DOS and Windows. Further, a well-known graphical user interface (GUI) for the visual operation and representation of an Object allocation table (OAT) is the standard Microsoft Windows Explorer.

Object allocation tables (OAT)'s are essential tools for the addressing of objects because, in order to address a data-object, a unique location reference and a unique identifier is needed; i.e. a *one-to-one* relationship between identifier and object. However, the application of Object allocation table (OAT)'s for handling objects by means of a graphical user interface (GUI) leaves a lot to be desired. The one-to-one relationship between identifiers and objects entails that an object can have only one identifying feature.

Deployment of a hierarchical structure of Object allocation tables (OAT)'s and,

hence, of object identifiers, indeed allows users to make an educated guess about the contextual identity of objects by following a certain 'path' of identifiers. Yet it does not allow users to describe and select objects based on their full identity. A full identity consists of many identifying features, each of which may be of interest to a user.

A system in accordance with one embodiment of the present invention contains a system that enables many-to-many connections between attributes and objects. In this way the full identity of an object can be described in terms of attributes and objects can consequently be addressed and controlled via any attribute, or combination of attributes, that has been connected to it.

One application of the above-mentioned system is to couple filenames (objects) from a file system to descriptive indexes (attributes). This enables the user to 'store' a filename under a number of indexes (attributes) describing, for instance, the file's content. The user may attach, for example, the descriptive indexes 'offer', 'John' and 'XYZ, Inc.' to a filename. Advantageously, the user can now retrieve this file via each index or via a combination of these. This embodiment of the invention is hereinafter referred to as the polymorphous object allocation table (POAT) system.

The polymorphous object allocation table (POAT) system can be stored on any computer system, and implemented at any level, for example as a file system in itself or as a shell on top of an existing file system.

Fig. 20 illustrates exemplary arrays employed for cross-referencing various attributes and objects in accordance with one embodiment of the invention.

With reference to Fig. 20, in accordance with one embodiment of the invention, the polymorphous object allocation table (POAT) system 2008 consists of two

object allocation tables 2014 and 2010 also referred to as OAT 1 and OAT 2 respectively and a two column table 2012 referred to as polymorphous attribute connection table (PACT) as described below.

Object allocation tables 2014 and 2010 are coupled by polymorphous attribute connection table (PACT) 2012. Object allocation table 2014 contains attribute identifiers A, B, C, D and E. Object allocation table 2010 contains object identifiers 1, 2, 3, 4 and 5. The polymorphous attribute connection table (PACT) contains two columns X and Y.

When a user desires a connection between an object and an attribute, a chosen object identifier in table 2010 is duplicated in polymorphous attribute connection table (PACT) under column 'x' in table 2012. Column 'y' from table 2012 holds a duplicate of a chosen attribute identifier, represented in table 2014. Thus, for example, a filename represented by object identifier '1' in table 2010 can be connected to attribute 'B' in table 2014. Integers in the same row of the polymorphous attribute connection table (PACT) form a unique connection between object and attribute and are referred to as object-attribute pairs.

In the way described above, the same object can be connected to many attributes. Connecting the same object identifier in object allocation table 2010 (OAT 2) to different attribute identifiers in object allocation table 2014 (OAT1), each time forms a new and unique object-attribute pair. In fact the invention allows the user to connect an unlimited number of attribute identifiers to a single object identifier (many-to-one), something that is not possible when using a prior art object allocation table (OAT) table, as described before. When used in the opposite way, the invention allows an unlimited number of object

identifiers to be connected to *one* attribute identifier (one-to-many).

According to another embodiment of the invention, the polymorphous object allocation table POAT may be structured to control the access of individual users, such as in network environments. User access to identifiers is granted via a structure of object allocation tables object allocation table (OAT)'s and polymorphous connection tables (PCT's), as described hereinafter.

In Fig. 21 each individual user is given a unique identifying integer in a user-OAT table 2102. Furthermore, groups of users are also given a unique identifying integer in a user group-OAT table 2104. Membership of an individual user to a user-group is granted by coupling a user-integer with a user group-integer in a 'membership'-PCT table 2106. The coupled integers generate a unique identifier integer known as the 'security-ID' 2112. At the moment a user is created, a special group is created in table 2104 with a corresponding link in 'membership'-PCT table 2106. This way it is ensured that each user has at least one security-ID. Security-ID 2112, in turn, is coupled in a 'rights'-PCT table 2108 to object identifier-integers from object allocation table (OAT 1), which contains attribute identifiers as previously discussed in connection with Fig. 20. The coupled identifiers in the 'rights'-PCT can be supplemented with a large number of 'rights' attributes contained in columns 2114, 2116, etc. Such 'rights' attributes can, for instance, determine what actions individual users may perform on the objects, like reading, writing, renaming or deleting.

As a result, different users - group combinations may have various access authorizations to attributes provided in a file management system in accordance with various embodiments of the present invention.

According to yet another embodiment of this invention, the aforementioned system, as described, for example, in Figs 1a and 1b, can be implemented as a program or shell user interface to enable users to easily or more efficiently create associations with data. With reference to FIG. 1c, a computer interface 100 for a document indexing system is shown, according to such an embodiment of the invention. Computer interface 100 comprises a right pane 102, a left pane 104, a menu bar 106, and a status bar 108. Right pane 102, referred to as an attribute view, displays a button bar 110, a plurality of rounded folder icons 112, a plurality of link icons 126, and a plurality of attribute icons. Furthermore, left pane 104, referred to as a file view, displays three tabs, a browse tab 116, an unindexed tab 118, and a retrieved tab 120, each having a different display when selected, as will be explained below. According to this embodiment of the invention, illustrated in FIG. 1c, attribute icons 114 comprise light icons 126 that operate as link icons to associate and disassociate attributes with one or more objects. As will be explained in more detail, the links and attributes each have visual cues to facilitate the indexing process. Also, it is understood that the objects, links, and attributes may be graphically or non-graphically represented in many forms other than as described herein.

To facilitate the indexing and retrieval in this system, according to one embodiment of the invention, the computer interface is divided into two windows or panes. The right pane is referred to as an attribute view since it is the location where attributes are created, removed, or otherwise manipulated. The left pane is referred to as an object view since it is the location where all indexed and unindexed documents or files can be viewed, selected, copied,

moved, etc. The structure and function of these two panes are hereinafter described.

Object view 104 is configured to display object icons, such as document files 122, in response to user input or selection, among other things. These object icons represent, for example, actual documents or other data stored on a computer system and are displayed in such a manner in order to facilitate the indexing, retrieving, and browsing functions of the system in accordance with one embodiment of this invention.

Depending on which of the three aforementioned tabs 116, 118 and 120 are selected, a different arrangement of files or icons are shown in the object view. Specifically, with continued reference to Fig. 1c, the retrieved tab is shown selected, whereas object view 104 displays document files 122, corresponding to a particular search result. As will be described in more detail with reference to the attribute view, the retrieved tab displays files that have been found by the system in accordance with one embodiment of the invention during a user-initiated search action. Typically, these displayed files have been previously indexed, linked, or associated to one or more attribute icons. In response to a user selecting the corresponding attribute icon these files are displayed within the area defined by the retrieved tab. Notably, all linked files are displayed together regardless of the directory they are physically stored in. In addition to retrieving indexed files, users can use this tab-view to modify or create associations as well.

Selecting the browse tab would provide users with a directory structure for a familiar way to browse directories, allowing them, at the same time, to index files. In this context, indexing a file corresponds to the tasks relating to setting

an association between an attribute represented by an attribute icon and a file represented by an object icon in the object view. This is the only tab that allows users to browse directories. This view is advantageously designed to give users an integrated standard way to navigate towards files. In one embodiment, for example, the browse tab may provide a directory structure similar to the Explorer interface of a Microsoft Windows operating system. An example of such a display, as provided by the browse tab, is shown with reference to Fig. 2 in object view 104. Advantageously, this enables the program to dynamically update associations so that documents can be copied and moved within the program and their associations will be maintained.

Fig. 3 illustrates the view of the unindexed tab on object view 104. This tab displays certain files that have not yet been indexed, or have not been indexed sufficiently, and is useful for enabling users to efficiently associate attributes without complicated searching. Notably, the user may customize the unindexed tab to only display files having a specific criteria. Such criteria include, for instance, at what locations to search for documents, the age and type of searched documents and the number of attributes already associated with searched documents. For example if the user only wants to index Microsoft Word documents created in the past seven days, he or she may instruct the program to only list the unindexed files having those criteria.

In addition, these displayed files from each tab have typical interface properties, similar to other graphical applications, including but not limited to the ability to be dragged to another window, copied, deleted, renamed, moved, etc. Also, double clicking a file opens the corresponding computer document in its proper application. For example, double clicking a particular word

processor file or icon could open that document in Microsoft Word. Similarly, a MIDI document could be opened and played in Microsoft Media Player.

With further reference to Fig. 1c, attribute view 102, is the area where link icons 126 and attribute icons 114, collectively referred to as attribute buttons, are created and managed, in accordance with one embodiment of the invention. Essentially, each attribute button represents one attribute, as described above, which comprises zero or more links with documents or other objects. These links allow efficient storage and retrieval of such documents since only one click is needed to subsequently retrieve them, regardless of their physical location, as will be described in more detail. Understandably, attribute icons 114 can display text so that a name or other description can be associated with each one.

Each attribute button has two portions, corresponding to two functions. First, link icon 126, on the smaller left portion of the attribute button, shown here as a light icon, is used to create and to remove associations between attributes and selected objects. As explained above, these link icons also provide a visual indication to the user of such creation and removal of an association. Second, the right larger portion of the attribute button, generally referred to as an attribute icon 114, is used to retrieve previously indexed documents.

Basically, the creation of a link is facilitated by a two step process. First, the desired objects from the object view are found and selected. Second, one or more attribute buttons are activated via link icons 126, located on the left portion of each attribute button. Such activation causes all selected files to be linked with the corresponding attribute button or buttons. Preferably, a visual cue, such as change of color, is provided to signal that the corresponding

attribute button has been activated, referred to as an illuminated cue. Subsequent retrieval of any linked documents is as simple as clicking the corresponding attribute icon 114, whereby the object view displays the retrieved tab with all files having that link. After a link is created it remains until removed by re-selecting the appropriate objects and de-activating link icons 126, where the visual cue is now referred to as a non-illuminated cue. The link icons, located on each of the attribute buttons, preferably comprise additional functions to further facilitate the indexing process. Specifically, the link icons have, according to one embodiment of the invention, additional phases or visual cues. For example, with reference to Fig. 11, the link icon portion 1100 does not display a light. This inactive cue occurs when there are no selected objects, and hence nothing to index via the link icon.

Another phase or visual cue occurs when more than one object icon is selected each having different links, referred to as a partially illuminated cue. In this situation, it becomes difficult to indicate the associations between attributes and objects since no one indication accurately reflects the links or associations of all selected object icons. Therefore, a different icon, for example a partially lit or shaded link icon, is displayed to signify such a status, as is depicted in figure 17. Should the user subsequently select or click this shaded icon, all of the selected objects would become indexed, as is shown in figure 17, step 49, where all selected objects in the object view are associated with the selected link icon (black) in the attribute view. This situation is similar to the aforementioned illuminated cue. Similarly, if the user clicks the shaded icon a second time, all of the selected objects would become unindexed, as is shown figure 17, step 51, the selected objects in the object view become disassociated

from the link icon (white) in the attribute view. This situation is similar to the aforementioned non-illuminated cue.

In figure 1c, the attribute view further comprises folder icons 112 for potentially organizing the attribute buttons according to a multi-leveled structure. Similar to a hierarchical file system, attribute buttons can be associated with a folder or sub-folder and efficiently stored as necessary. Fig. 1c illustrates the use of folder icons 112. Moreover, in addition to displaying text, each folder icon contains a visual cue, such as an arrow 199, for signaling whether the folder is open or closed. According to this embodiment, a down-facing arrow signifies an open folder and a side-facing arrow signifies a closed folder. An open folder icon displays all attribute buttons or sub-folder icons associated with that folder, whereby a closed folder icon hides all of its contents.

Accordingly, one efficient use of folder icons is to avoid the clutter of countless attribute buttons by only opening the minimum number of folders needed. Selecting or clicking the folder icon is one method of interchangeably opening or closing it. Another effective use of folders icons is that it permits a contextual navigation through the attribute view, without having to inspect each attribute individually.

With regard to these folder icons, arrows 199 advantageously change color according to selections that exist within the folder. One type of behavior regards the selection of attribute icons. When, within a certain folder, one or more attribute icons are selected, the folder icon shows this by displaying, for example, a red outline around its icon, regardless of whether the folder is opened or closed. Another type of behavior regards the selection of link icons.

When, with a particular folder, one or more link icons are selected, this is shown by displaying the folder icon in, for example, a yellow color. If a certain folder contains selected link icons as well as selected attribute icons, both behaviors accumulate, that is, the folder icon would show, for example, yellow with, for example, a red outline.

It is noted that the attribute buttons as well as the object icons only represent and facilitate the storing of associations, not the actual objects. Thus, each document can be indexed numerous times despite being stored in only one physical location. This enables users to retrieve files according to innumerable associations, overcoming several limitations from the previously mentioned traditional file systems. For example, a word document written in April 1997, relating to copyright law, belonging to client number 8765, and comprising the text of a copyrighted novel, could be indexed four times, by its date, area of law, client matter, and topic. Although the physical storage of that document will remain in one location, as with the prior art systems, four indexes can be created here, one for each of the above four attributes, for more efficient storage and retrieval. Thus, the user is no longer pressured to sort documents according to a specific directory system, since every association can be incorporated into the system of this invention.

The indexing process can be further manipulated via button or icon options on the button bar 110. As mentioned above, when multiple attribute icons are selected the retrieved tab only displays documents that are indexed with all of the selected attributes. Tighten icon 142 and loosen icon 140 are configured to modify this result. Specifically, if three attribute icons, for example, are selected the retrieved tab would normally only display documents indexed with

all three attributes. Display 141, on the button bar, would then state "3/3" to signify that only documents matching all three of the three selected attribute icons are retrieved. The user may optionally click or select loosening icon 140 to reduce the criteria to "2/3," whereby any document matching 2 ore more of the three attributes are displayed. The loosening icon can be clicked repeatedly, as low as "1/3," or all documents with at least one matching attribute. Similarly, tighten icon 142 can be clicked to reverse the effect of the loosening icon, up to a maximum of "3/3." Logically, these icons can be used regardless of the number of attribute icons selected.

Also positioned on button bar 110 are sort descending icon 144, show selected only icon 146, user sort icon 148, deselect indexes icon 150, and stop icon 152. Briefly, the sort descending icon 144 rearranges the folder and attribute icons in alphabetical order. The show selected only icon, when selected, displays only all attribute icons that are associated with a selected document or documents (i.e. all attribute buttons with link icons that are turned on). Alternatively, user sort icon 148, displays the attribute icons in the order that the user has chosen by dragging attribute icons to their appropriate places. Stop icon 152 stops any pending process, such as the retrieval of indexed documents. It is understood that additional icons that incorporate other features may be included as well.

The method of assigning attributes and retrieving documents, according to one embodiment of this invention, will now be explained with reference to FIGs. 4-10, which respectively illustrate a flow diagram and graphic illustrations of the relevant steps of the diagram.

Initially, at step 400 of Fig. 4, the desired document or documents are selected for indexing. For example, a user may select several accounting spreadsheet

files to be indexed together under one or more attributes, such as those titled 'accounting' or 'spreadsheets.' As described above, the object view has a browse tab which can be used to easily navigate a computer system directory to search for these desired documents. Alternatively, the unindexed tab can be used to check if the documents are listed there. Once the documents have been located they are then all selected or highlighted with a mouse or other device. Fig. 5 illustrates what one document, shown here as document 500, would look like when selected.

Thereafter, at step 402, one or more associations are created between the selected documents and the attribute buttons 114. This is simply accomplished by clicking and turning on the link icons contained in the attribute buttons. Fig. 6, illustrates link icon 126 being clicked, whereby the same link icon in Fig. 5 is shown prior to being clicked. Further, with reference to Fig. 7, link icon 126 is shown turned on, which signifies that the selected files or documents are presently linked. If the user clicks the link icon a second time the attribute or attributes would be removed from their association, and the light would turn "off."

Another illustration is shown with FIGs. 9 and 10, whereby the document titled "Charlie Chaplin" is indexed with attributes "Famous" and "Human" and the document titled "Eiffel tower" is indexed with attributes "Famous" and "Very High".

Notably, not only can multiple selected files be indexed simultaneously, but an entire directory, and its subdirectories, can be simultaneously indexed by simply selecting the directory icon and clicking the desired link icons.

Now that the attributes have been associated with certain documents, the user

may subsequently retrieve these documents, at step 404, by clicking or selecting one of the associated attribute icons. Fig. 8 shows, according to one embodiment of the invention, how an attribute icon appears when clicked or selected. Attribute icon 800 is shown before selection and attribute icon 802 is shown selected. After the attribute icon is selected, the text on the attribute icon changes color to signify its present selected status, whereby re-clicking the attribute icon would deselect it and return the color to its normal status. This process of selecting an attribute icon instantly causes the object view to display all files or documents associated with that attribute icon. Furthermore, the selection of multiple attribute icons is possible, whereby only those documents associated with all of the selected attribute icons will be retrieved. For example, with reference to the documents of FIGS. 9 and 10, the user can select the attribute icon "Famous" and retrieve both documents "Charlie Chaplin" and "Eiffel Tower." On the other hand, the user can simultaneously select attribute icons "Famous" and "Human," whereby only the document title "Charlie Chaplin" would be retrieved. This result occurs because only "Charlie Chaplin" is linked with both "Famous" and "Human." The user may also use the tighten icon 142 and/or loosen icon 140, as described above, to further manipulate these results.

At step 406, the user browses the retrieved tab in the object view to find the document he or she is searching for. Thereafter, the document can be double clicked to execute in its proper application, as detailed above. For example, if the previously mentioned document titled "Eiffel Tower" was a picture, the user could double-click it to execute the document in an application for viewing images. This step concludes the indexing and retrieving process.

This invention thus makes it possible for anyone to find a document quickly, even when the searching person has no prior knowledge of the existence of the document, nor its associated attributes. As an example, suppose that employee "A" has "sent" an "offer" to client "C" in "February" of "1998" and that the offer was about delivering product "X" and that it was an "important" offer that has been "sanctioned" by the "management" and that it is drafted in a "standard offer format." When everything between quotation marks has been indexed by the system, as described herein, the user will be able to retrieve the document in a number of ways. For instance, this document can be found by requesting all "offers" that have been "sanctioned" by the "management" in "1998", or by requesting all correspondences of employee "A" to client "C" in "February", or by requesting all documents in "standard offer format" drafted in "1998", etc. This invention retrieves every document associated with all of the attributes selected by the user.

The process by which this invention stores and maintains links will now be explained in more detail, according to one embodiment of the present invention. An link is essentially an association between an attribute represented by an attribute icon and one or more documents. One method used by this invention is a distinctive way of simplifying the index process. Simply, every attribute button and linked document is assigned a unique integer. When a particular attribute is linked to a particular document an entry is added to a Polymorphous attribute connection table (PACT) comprising both the integer of the attribute button and the integer of the document. Whenever the program needs to determine a particular link it looks at all the entries to see which files are linked to which attributes. To optimize this process the entries of the

Polymorphous attribute connection table (PACT) may be additionally sorted by attribute and/or document. Also, pointers may be used to point to all entries of a particular attribute for more speedy responses. Naturally, there are separate tables for the attributes and the documents to associate a particular integer with the appropriate document or attribute name, location, etc.

It is understood that the present invention is not limited to the embodiments as described hereinabove. The links and attributes can be combined as one portion of the attribute button, rather than two. This would be accomplished by using additional visual cues and utilizing the right and left button of a mouse, for example, to alternate between the functionality of the link and attribute aspect respectively. For example, the left mouse button can be used to associate an attribute with selected files and the right mouse button can be used to retrieve all previously indexed files. Furthermore, in an alternative embodiment the object and attribute views can be interchanged or one view may be larger than the other. Also, the object view could have multiple instances of a particular tab view. For example, the browse tab could display two instances for dragging or manipulating files between multiple locations. In addition, the interface can be modified to include additional tabs, panes, menu options or buttons. Also, the attribute buttons can be altered to appear as a tree structure.

Moreover, the unique indexing mechanism of this invention has many alternative applications in addition to storing and retrieving documents as explained hereinafter.

In some cases a stored object may have internal characteristics that may be interpreted as attributes by an embodiment of the invention and

presented to the user in an appropriate form. To this effect an area of the system's interface is reserved to display one or more of the attributes. For example, in an embodiment of the invention that manipulates uniform resource locators (URL's) the system treats a user selected URL in a way similar to an Internet browser application like Microsoft Internet Explorer or Netscape and displays the result in the reserved area of the system's interface. Additionally, these attributes can aid the user in determining the nature of the stored object in order to discriminate it from other stored objects with equal attributes. In an extension of the invention such attributes may be exploited as data fields for additional search criteria. Using common database interface techniques the user can define an SQL query on the attributes. The invention subsequently applies said query on the collection of retrieved objects. In an embodiment of the invention that manipulates files, the user may search the retrieved files for files with creation dates within a certain range. In another embodiment of the invention that manipulates electronic mail, the user may search the retrieved messages for messages with a specific line of characters (including wildcards) in the title or the body of the message. In yet another embodiment of the invention that manipulates electronic business cards, the user may search the retrieved cards for addresses within a specific state. In yet another embodiment of the invention that manipulates database records, the user may search the retrieved records for records complying to an SQL query. In yet another embodiment of the invention that manipulates images, the user may search the retrieved images for certain color values.

Fig. 22 illustrates some graphic examples of alternative embodiments of the invention. For example E-mail messages may be indexed in accordance with

the principles of the present invention. A user then selects a desired email message in object display area 2202, in response to which, title and content of the selected message is displayed in display area 2204. For graphics objects, for example, in response to a user selected object, display area 2204 may display attributes of the selected graphics object and its graphical representation. For database objects, in response to a user selected record in display area 2202, fields of the selected record are displayed in display area 2204. Similarly for business card objects, in response to a user selected card in display area 2202, data and the picture of the owner of the card may be displayed in area 2204. With reference to URL objects, in response to a user selected URL in display area 2202, the contents and presentation of the selected URL may be displayed in display area 2204.

In accordance with another embodiment of the invention, the system can be used to communicate the content of Internet locations. In this embodiment, a pane such as pane 102 in figure 2 is used to display attributes. This pane can display the user's custom attributes, but it can also be directed to a particular Internet location. This Internet location is asked to send the attributes with which the contents of that location was registered. The pane then displays those obtained attributes. For example, when the Internet location would be 'www.XYZ.com,' the XYZ server would be inquired about its attributes, and a pane such as 102 would display the content of the XYZ web site in the form of representative attributes. The user could then click one or more attributes according to the nature of the information required. The invention then queries the server or Internet location with the set of selected attributes. The server sends back an HTML-page with results, for example, all URLs within the

location that match the selected attributes. The result would then be displayed in a viewer window. A normal browser such as Microsoft Internet Explorer or Netscape Communicator could also be used for such a purpose.

When used in an organization environment, multi user features, security and other appropriate features can be applied. By default, attributes are divided in two separate groups. The first group of attributes is visible to all participating users, the second is visible to the individual user only. This enables a user to apply his/her private attributes as well as the company's attributes. In addition to this, an embodiment of the invention makes it possible to give a user different kinds of access rights to particular groups of attributes, and not to others. In this way, a management department for example can use different, or partly different, attributes than the sales department. Users can be permitted read access, read/write access and delete access to particular indexes, thereby restricting the creation of new attributes to specific users, such as system administrators. This gives the organization a chance to keep a steady list of attributes that all participants are familiar with.

The attribute system of this invention can also be customized to act according to a predetermined set of rules. For example, all Microsoft Word documents could automatically be indexed with certain default attributes. Further, the name of the document or first line of text could signal what attributes should be associated with it by the program.

Thus, while there have been shown and described and pointed out fundamental novel features of the invention as applied to alternative embodiments thereof, it will be understood that various omissions and substitutions and changes in the form and details of the disclosed invention

may be made by those skilled in the art without departing from the spirit of the invention. It is the intention, therefore, to be limited only as indicated by the scope of the claims appended hereto. It is to be understood that the drawings are not necessarily drawn to scale, but that they are merely conceptual in nature.

CLAIMS

1. A method for displaying an interactive user interface for associating objects represented by object icons to one or more of a plurality of attributes represented by attribute icons having corresponding link icons, said method comprising the steps of:

selecting one or more of said object icons;

selecting one or more of said link icons to create an association between said selected object icons and attributes represented by attribute icons corresponding to said selected link icons; and

providing a visual indication of said association by modifying the appearance of said selected link icons and said selected object icons.

2. The method in accordance with claim 1 wherein said step of selecting further comprises the step of moving a cursor pointer over a desired object icon and providing an indication of said selection.

3. The method in accordance with claim 2 wherein said step of moving a cursor pointer further comprises the step of moving a computer mouse so as to control the movement of said cursor pointer and said step of providing an indication of said selection further comprises the step of clicking

on a first control button of said computer mouse.

4. The method in accordance with claim 3 wherein said step of selecting a link icon further comprises the step of moving a cursor pointer over said link icon and providing an indication of said selection.

5. The method in accordance with claim 4 wherein said step of moving a cursor pointer over said link icon further comprises the step of moving a computer mouse so as to control the movement of said cursor pointer and said step of providing an indication of said selection further comprises the step of clicking on a first control button of said computer mouse.

6. The method in accordance with claim 5 wherein said step of modifying the appearance of said link icon further comprises the step of changing the color of said icon.

7. The method in accordance with claim 6 further comprising the step of removing said association between an attribute icon and an object icon by selecting a corresponding link icon and providing an indication signal corresponding to said removal of association.

8. The method in accordance with claim 7 wherein said step of providing an indication signal corresponding to said removal of association further comprises the step of moving a cursor pointer by a computer mouse over said corresponding link icon and clicking over said modified link icon.

9. The method in accordance with claim 8 further comprising the step of retrieving said objects represented by object icons by selecting at least one of said attribute icons and displaying all object icons that have been previously associated with said selected attribute icons.

10. The method in accordance with claim 1 further comprising the step of displaying said link icons and said attribute icons to a plurality of user connections in a computer network.

11. The method in accordance with claim 10 further comprising the step of displaying a selective set of link icons and attribute icons on each of said user connections based on a security level.

12. An interactive user interface comprising:

a plurality of objects represented by object icons displayed on a computer display device, said object icons being selectable by manipulating a user interface device;

a plurality of attributes represented by attribute icons having corresponding link icons displayed on said display device, said attribute icons and said link icons being selectable by manipulating said user interface device wherein a selected link icon exhibits a modified appearance so as to indicate an association between a corresponding attribute icon and a selected object icon;

and

a polymorphous attribute connection table for tracking associations established between said attribute icons and said object icons.

13. The user interface in accordance with claim 12 wherein said interface device provides instruction signals for moving a cursor pointer over a desired object icon and provides an indication signal when a user desires to select said desired object icon.

14. The interface in accordance with claim 13, wherein said user interface device is a computer mouse.

15. The interface in accordance with claim 12 wherein said modified appearance of said link icon further comprises a color change.

16. A method of associating data files represented by object icons on a display device, to one or more of a plurality of attributes represented by attribute icons having corresponding link icons displayed on said display device for efficient indexing and retrieval of said data files on a computer storage device, comprising the steps of:

(a) selecting one or more of said data files;

(b) selecting one or more of said link icons to create an association

between said selected data files and said attributes represented by said attribute icons corresponding to said selected link icons; and

(c) providing a visual indication of said association by modifying the appearance of said selected link icons.

17. The method in accordance with claim 16 wherein said step of selecting one or more of said data files further comprises the step of moving a cursor pointer over a desired one or more of said object icons and providing an indication of said selection.

18. The method in accordance with claim 17 wherein said step of moving a cursor pointer further comprises the step of moving a computer mouse so as to control the movement of said cursor pointer and said step of providing an indication of said selection further comprises the step of clicking on a first control button of said computer mouse.

19. The method in accordance with claim 18 wherein said step of selecting said link icons further comprises the step of moving a cursor pointer over each of said link icons and providing an indication of said selection.

20. The method in accordance with claim 19 wherein said step of moving a cursor pointer over each of said link icons further comprises the step of moving a computer mouse so as to control the movement of said cursor

pointer and said step of providing an indication of said selection further comprises the step of clicking on a first control button of said computer mouse.

21. The method in accordance with claim 19 wherein said step of modifying the appearance of said link icons further comprises the step of changing the color of said link icons.

22. The method in accordance with claim 21 further comprising the step of removing said association between an attribute icon and a file object icon by selecting a corresponding link icon and providing an indication signal corresponding to said removal of association.

23. The method in accordance with claim 22 wherein said step of providing an indication signal corresponding to said removal of association further comprises the step of moving a cursor pointer by a computer mouse over said corresponding link icon and clicking over said modified link icon.

24. The method in accordance with claim 23 further comprising the step of retrieving said data files represented by object icon by selecting at least one of said attribute icons and displaying all object icon that have been previously associated with said selected attribute icons.

25. The method in accordance with claim 16 further comprising the step of displaying one or more folder icons, said folder icons corresponding to a desired group of said attribute icons.

26. The method in accordance with claim 25 further comprising the step of associating additional folder icons to previously displayed folder icons.

27. The method in accordance with claim 26 further comprising the step of displaying said attribute icons and said additional folder icons corresponding with a folder icon when said folder icon is selected and hiding said attribute icons and said additional folder icons when said folder icon is not selected.

28. The method in accordance with claim 24 further comprising the step of changing the appearance of said selected attribute icons.

29. A computer interface for a file indexing and retrieval system, comprising:

a first pane having a retrieved tab and a browse tab;

a second pane adjacent to said first pane;

a plurality of data files represented by object icon displayed on a computer display device, when said browse tab has been selected, said object icons being selectable by manipulating a user interface device;

a plurality of indexes represented by attribute icons having corresponding link icons displayed together on said second pane, said attribute icons and said link icons being selectable by manipulating said user interface device wherein a selected link icon exhibits a modified appearance so as to indicate an association between a corresponding attribute icons and a selected object icon; and

a polymorphous attribute connection table for tracking associations established between said attribute icons and said object icons.

30. The computer interface in accordance with claim 29, wherein each of said link icons displays one of the following four visual cues:

blank cue when none of said object icons are selected;

illuminated cue when said selected object icons are all linked with said attribute icons;

non-illuminated cue when none of said selected object icons are linked with said attribute icons; and

partially illuminated cue when a proper subset of said selected files icons are linked with said attribute icons.

31. The computer interface of claim 29, wherein said link

icons comprise graphic images that change position or color when selected.

32. The computer interface of claim 29, wherein said second pane further comprises a loosen icon and a tighten icon.

33. The computer interface of claim 29, wherein said first pane further comprises an unindexed tab for displaying a predetermined set of files.

34. The computer interface of claim 29, wherein said second pane further comprises one or more folder icons.

35. The computer interface of claim 29, wherein said first pane and said second pane are equally sized and positioned adjacently on left and right sides of said display device.

36. The computer interface of claim 29, wherein said first pane and said second pane can each be adjusted to increase or decrease in size and cover or decrease the space of the other.

37. The computer interface of claim 34, wherein said plurality of link icons and said folder icons are arranged in a tree structure format.

38. The computer interface of claim 29, further comprising a show selected only icon, upon activation of which only attribute icons

corresponding to a selected object icon are displayed.

39. The computer interface of claim 29 further comprising one or more supplemental icons corresponding to each of said attribute icons for performing boolean searches.

40. The computer interface of claim 39, wherein one of said supplemental icons is a filter icon.

41. The computer interface of claim 29 further comprising a third pane for displaying extended information on a selected one or more of said data files.

42. The computer interface of claim 41, wherein said third pane displays Internet WebPages, e-mail messages, graphics images, documents, videos, database records and electronic business cards.

43. In a computer system employed to retrieve a plurality of data files based on corresponding attributes of said data files, a polymorphous computer file system comprising:

a polymorphous attribute connection table (PACT) configured to store a plurality of object attribute pairs, wherein each of said pairs defines a relationship corresponding to a data file and an association of attributes; and

a processing means employing said POAT so as to retrieve data files having an association with one or more attributes selected for said retrieval.

44. The polymorphous computer file system of claim 43 further comprising:

a first object allocation table (OAT) having a plurality of attribute fields, each of said attribute fields corresponding to a unique one of said attributes; and

a second object allocation table (OAT) having a plurality of data file fields, each of said data file fields corresponding to a unique one of said data files.

45. The polymorphous computer file system of claim 44, wherein each of said object attribute pairs in said PACT have a first link to one of said attribute fields from said first object allocation table and a second link to one of said data file fields from said second object allocation table.

46. A method for storing data files on a polymorphous object allocation table (POAT) file system comprising the steps of:

selecting one or more data files;

associating one or more of said data files with one or more attributes;

recording said association on said polymorphous object allocation table (POAT) file system by adding one or more entries to a polymorphous attribute connection table (PACT) from a polymorphous object allocation table (POAT).

47. The method of claim 46 further comprising a method for retrieving said data files comprising the steps of:

selecting at least one of said one or more attributes; and

retrieving said one or more data files associated with said attributes by examining said recorded entries in said polymorphous attribute connection table (PACT).

48. The method of claim 46 further comprising a method for deleting said associations comprising the step of:

removing said association by deleting a corresponding entry from said one or more entries on said polymorphous attribute connection table (PACT).

49. The system according to claim 43 further comprising:

a multi-user network system for providing data file access to a plurality of users, wherein each user has a corresponding security level; and

a PCT rights table for storing said security level and associated life access rights corresponding to said security level.

50. The system of claim 49, wherein said file access rights attributes comprise restrictions selected from the group consisting of viewing said attributes, creating said associations, reading said data files, modifying said data files, modifying said attributes, renaming said files, deleting said files and deleting said attributes.

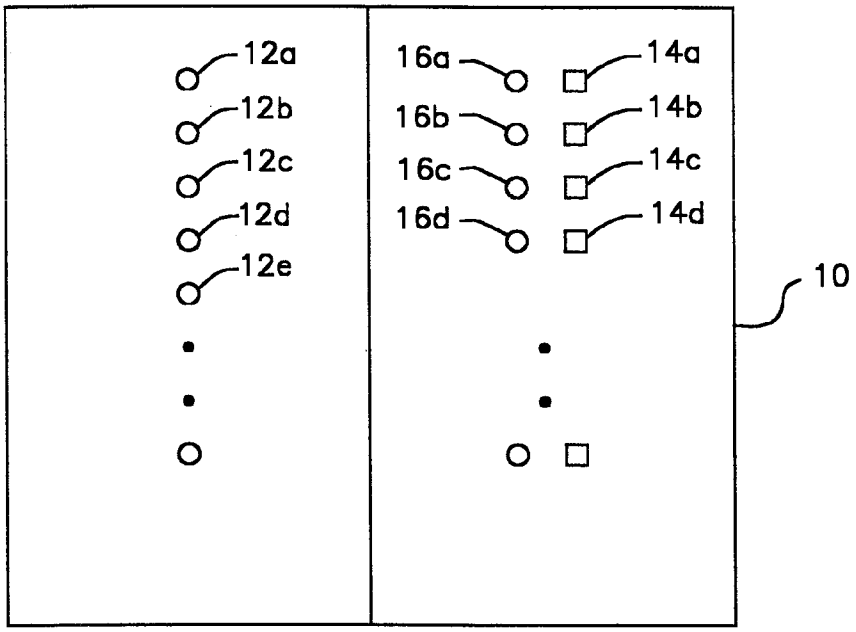


FIG. 1a

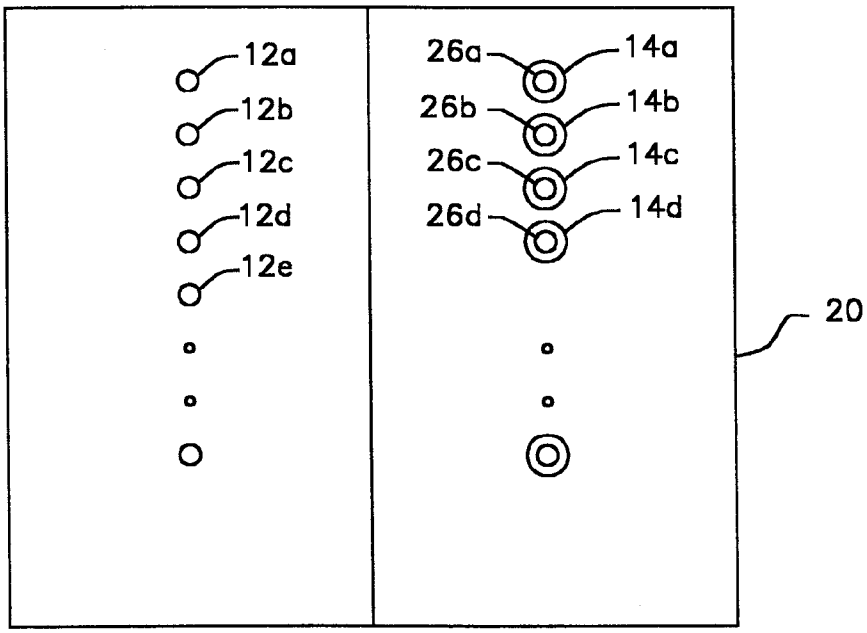


FIG. 1b

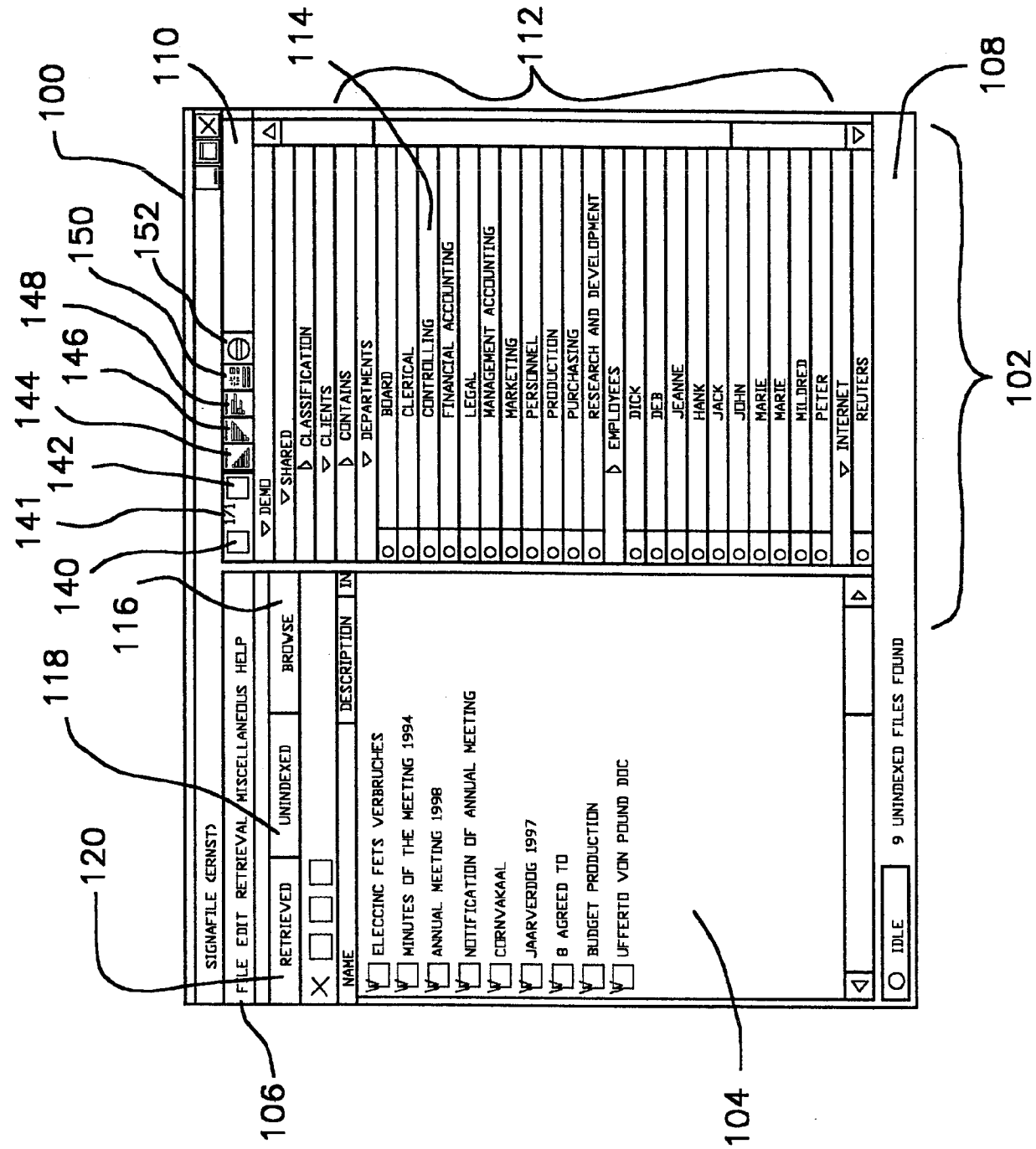
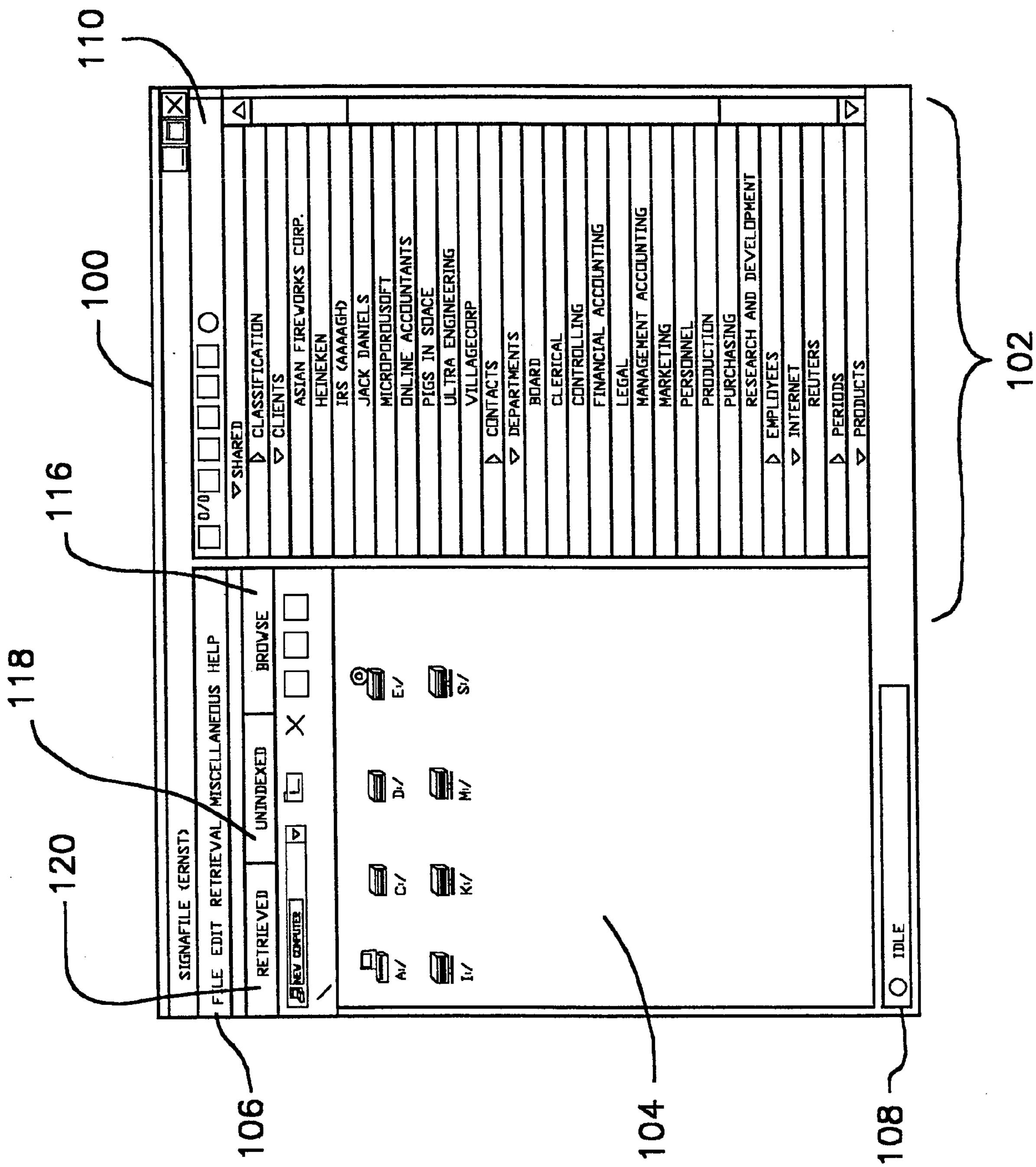


FIG. 1c



110
100
116
118
120
102
106
104
108
SIGNAFILE (ERNST)
FILE EDIT RETRIEVAL MISCELLANEOUS HELP
RETRIEVED
UNINDEXED
BROWSE
SHARED
CLASSIFICATION
CLIENTS
ASIAN FIREWORKS CORP.
HEINEKEN
IRS (AAAAGH)
JACK DANIELS
MICROPOROUSOFT
ONLINE ACCOUNTANTS
PIGS IN SPACE
ULTRA ENGINEERING
VILLAGECORP
CONTACTS
DEPARTMENTS
BOARD
CLERICAL
CONTROLLING
FINANCIAL ACCOUNTING
LEGAL
MANAGEMENT ACCOUNTING
MARKETING
PERSONNEL
PRODUCTION
PURCHASING
RESEARCH AND DEVELOPMENT
EMPLOYEES
INTERNET
REUTERS
PERIODS
PRODUCTS
A:/
C:/
D:/
E:/
I:/
K:/
M:/
S:/
IDLE

FIG. 2

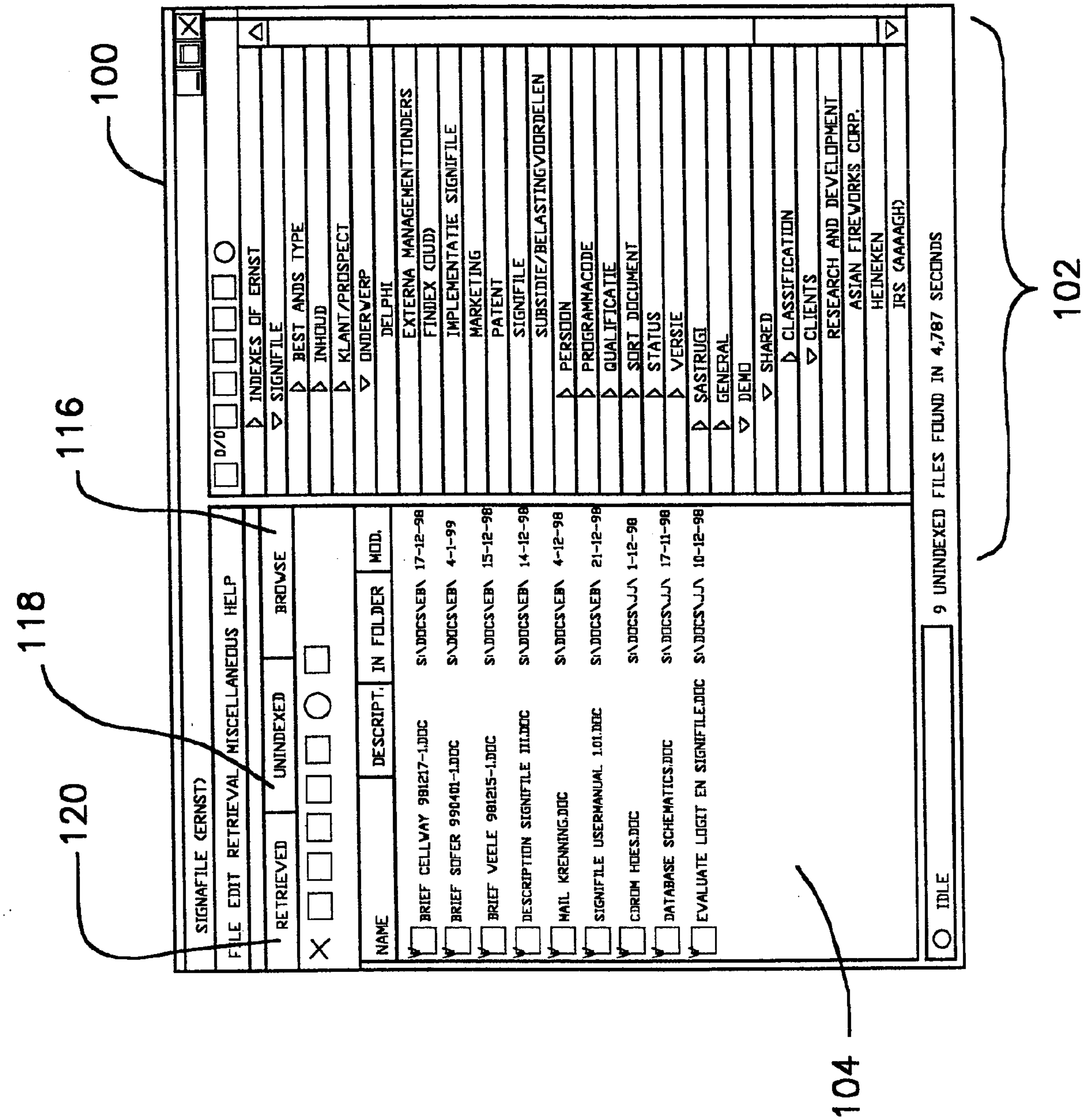


FIG. 3

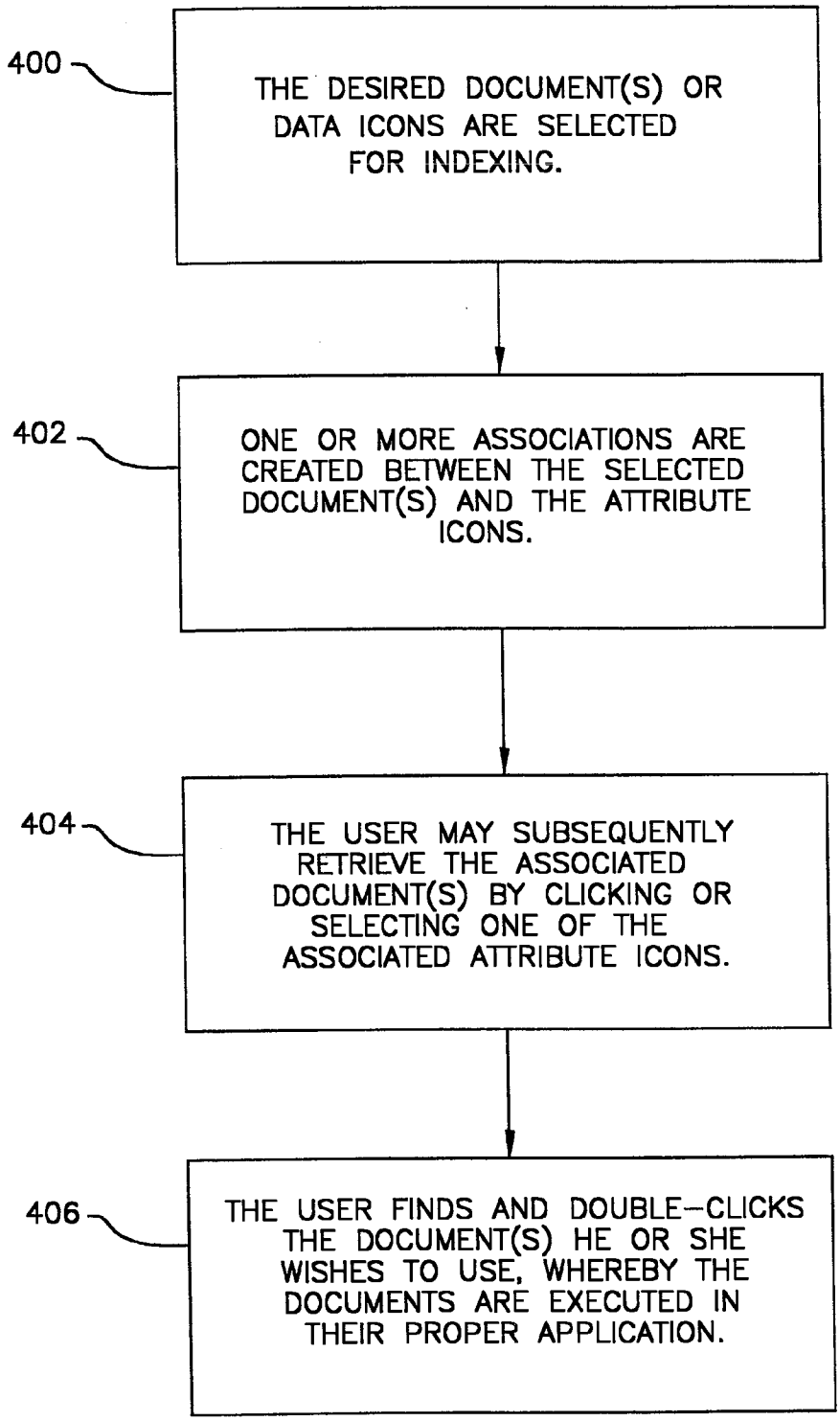


FIG. 4

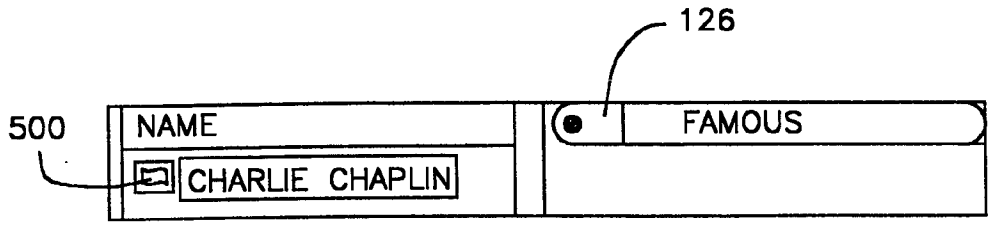


FIG. 5

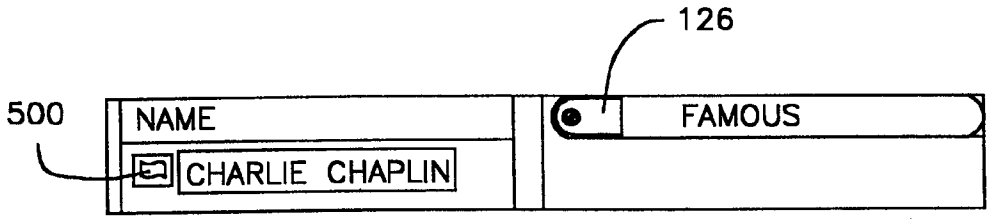


FIG. 6

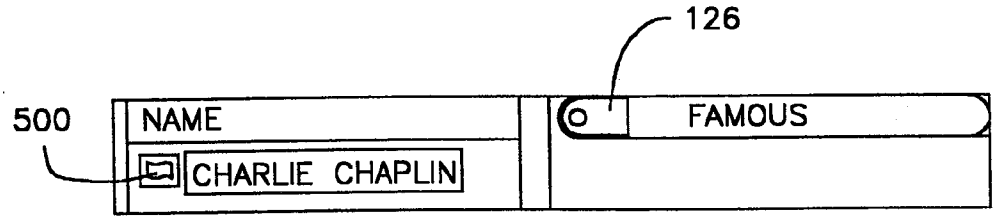


FIG. 7

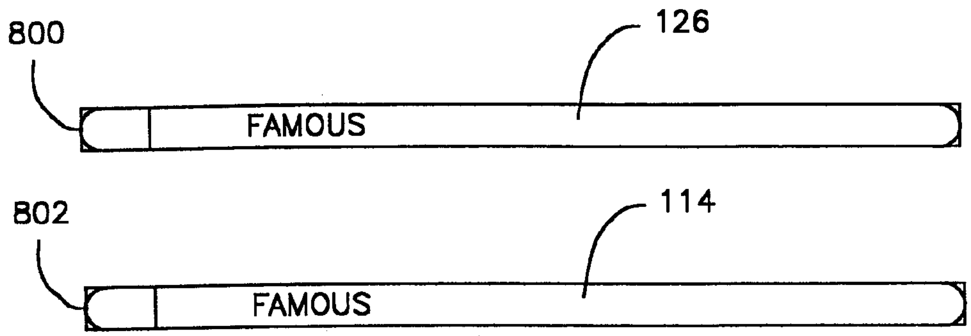


FIG. 8

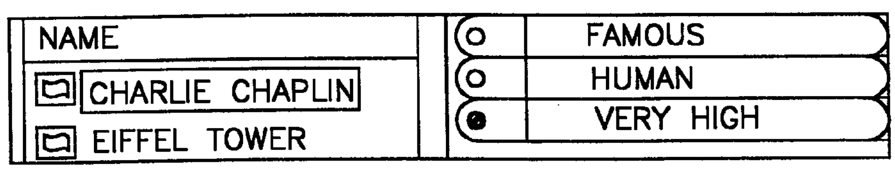


FIG. 9

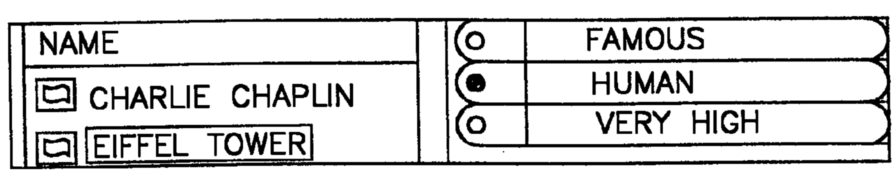


FIG. 10

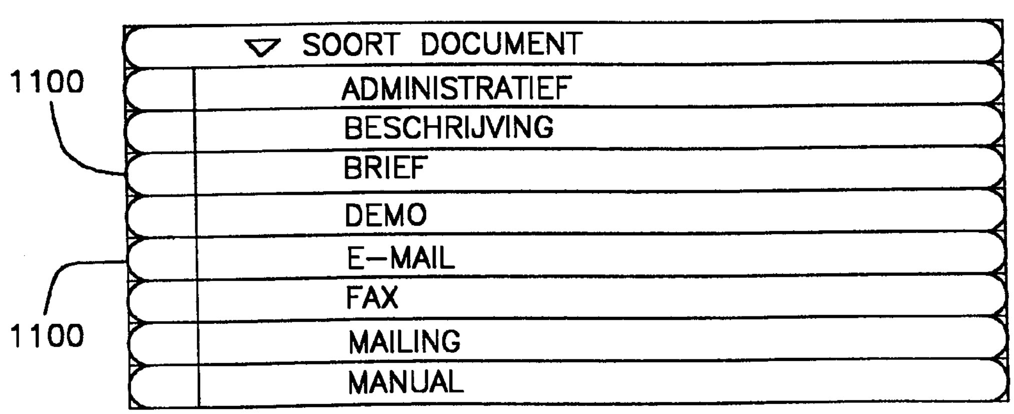


FIG. 11

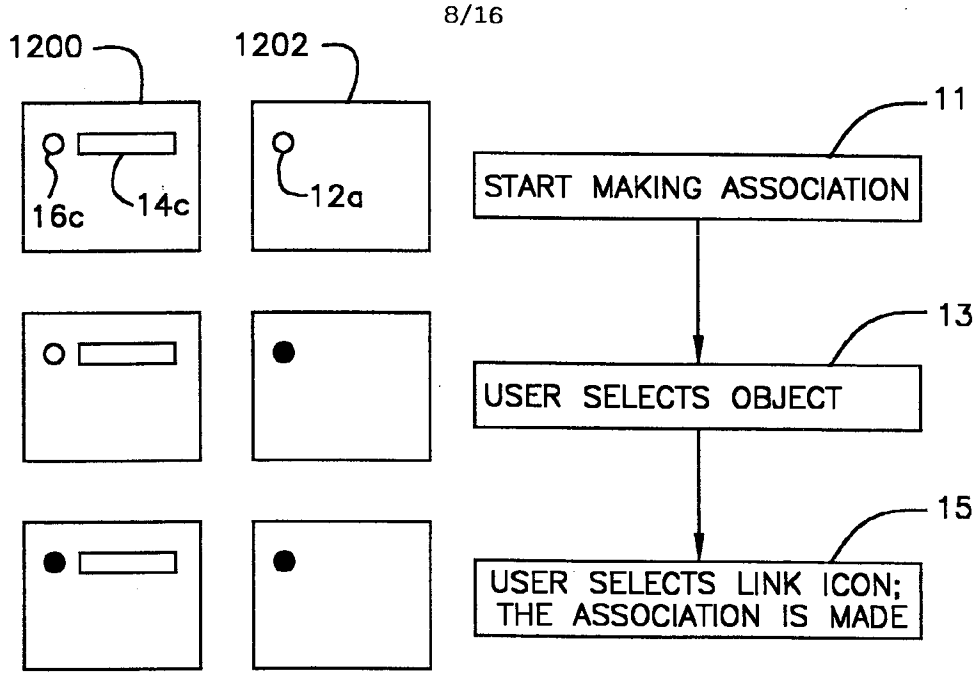


FIG. 12

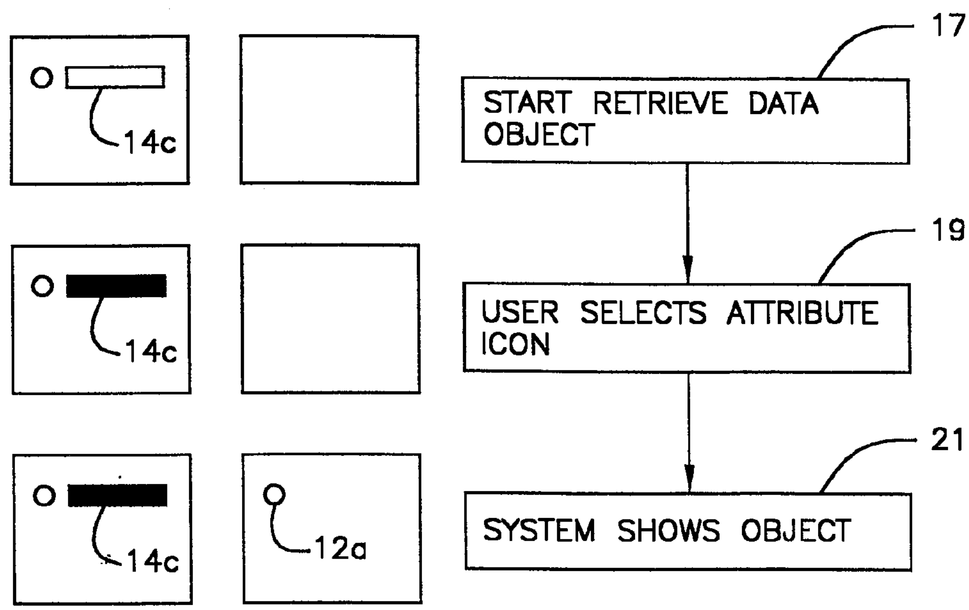


FIG. 13

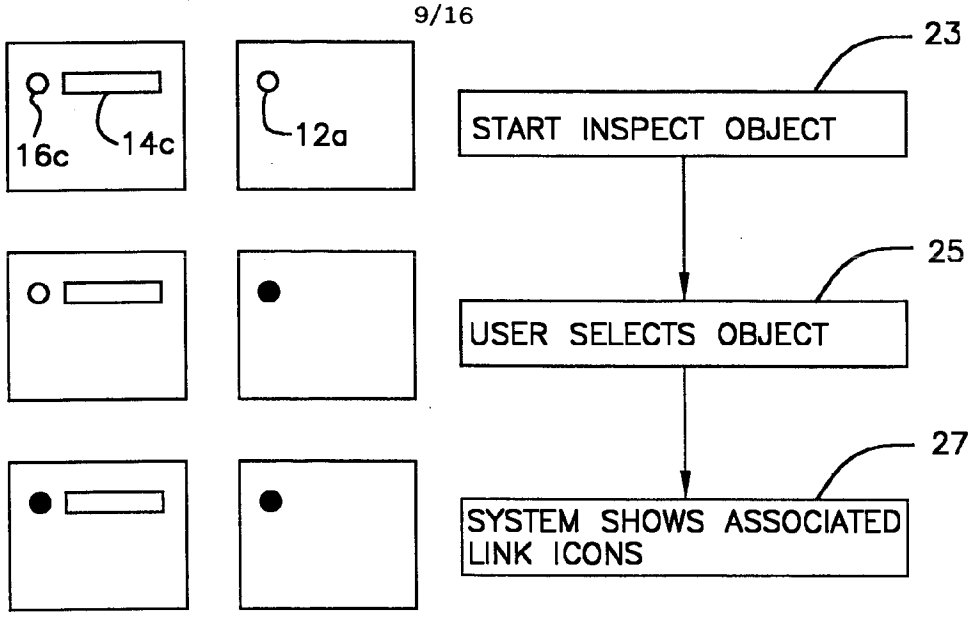


FIG. 14

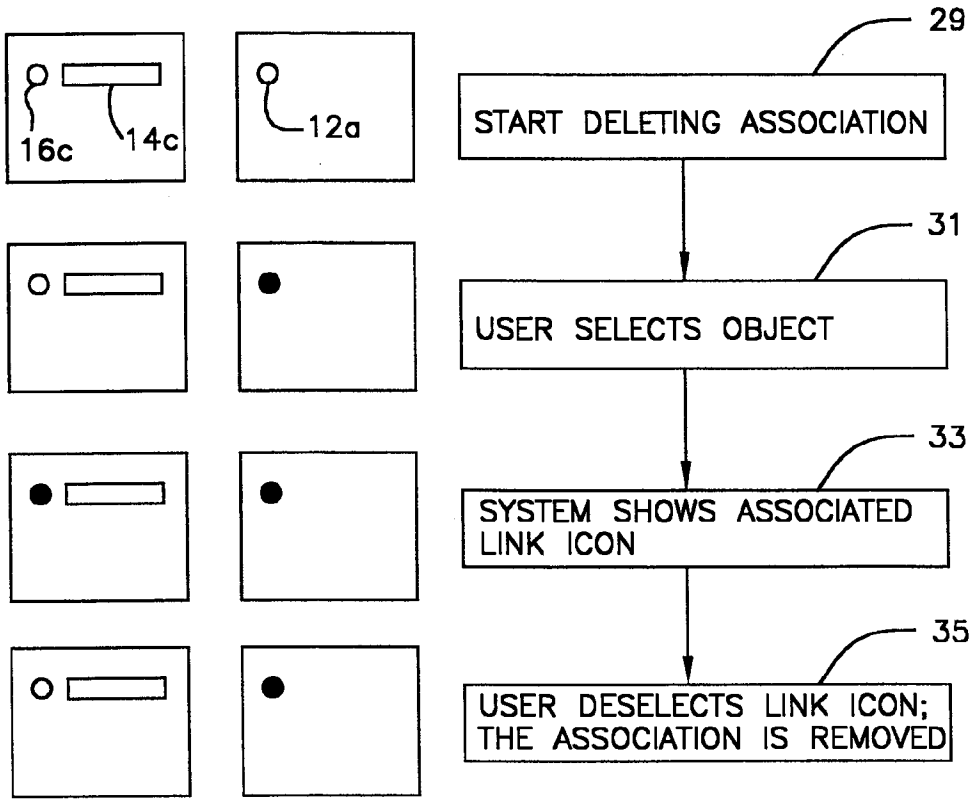


FIG. 15

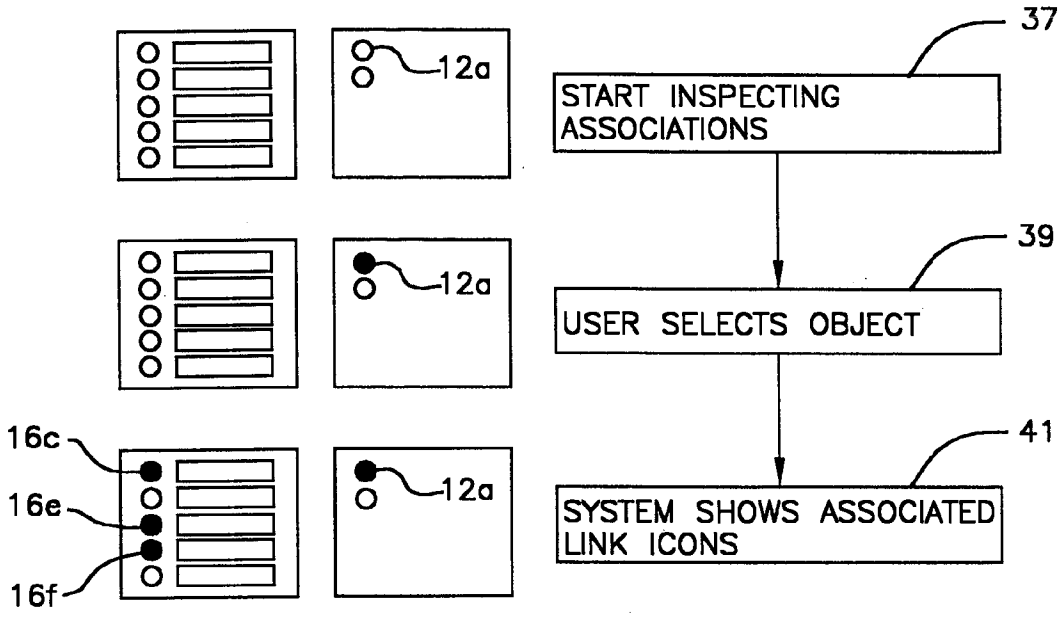


FIG. 16

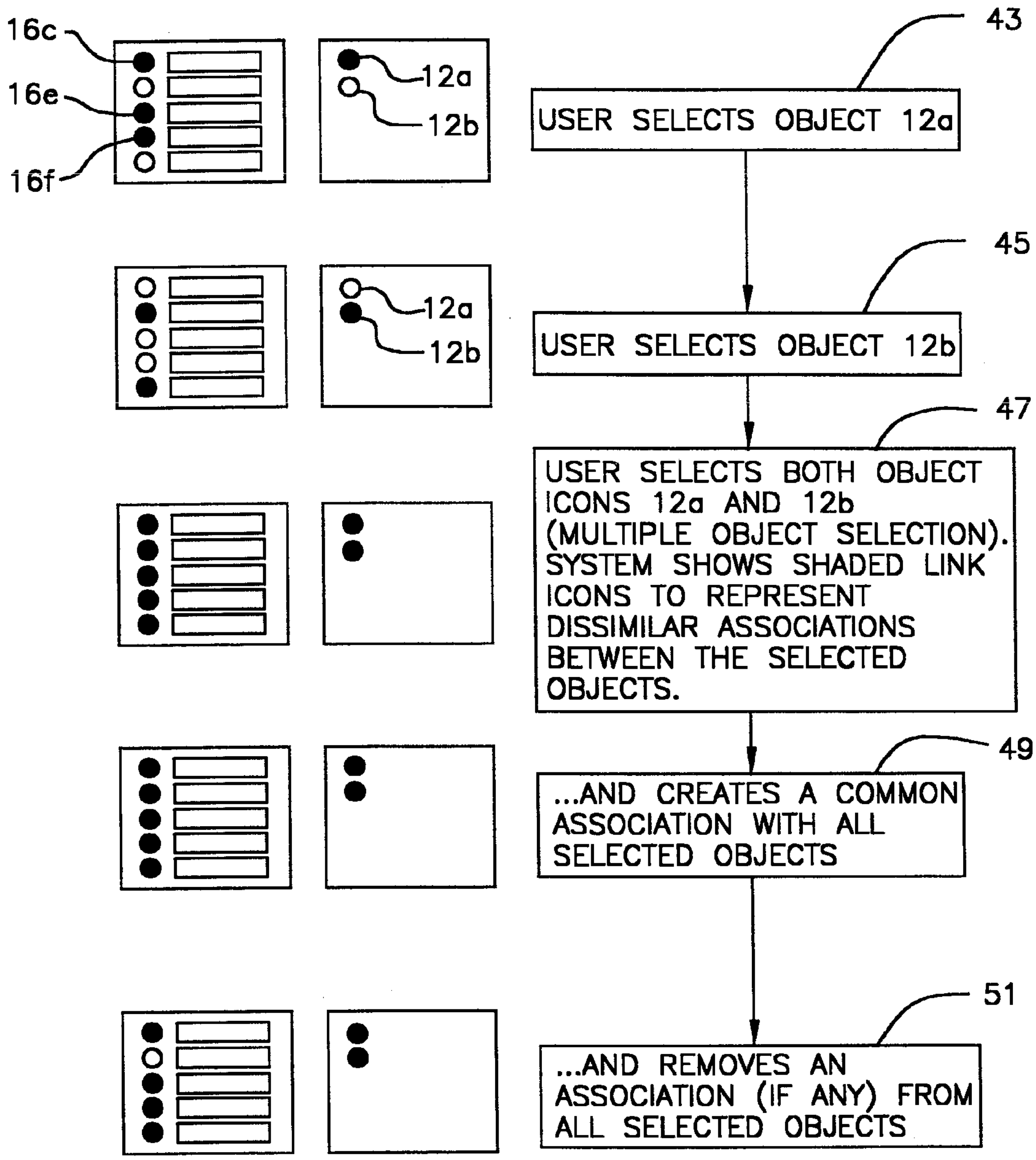

16c
16e
16f
12a
12b
43
USER SELECTS OBJECT 12a
45
12a
12b
USER SELECTS OBJECT 12b
47
USER SELECTS BOTH OBJECT ICONS 12a AND 12b (MULTIPLE OBJECT SELECTION). SYSTEM SHOWS SHADED LINK ICONS TO REPRESENT DISSIMILAR ASSOCIATIONS BETWEEN THE SELECTED OBJECTS.
49
...AND CREATES A COMMON ASSOCIATION WITH ALL SELECTED OBJECTS
51
...AND REMOVES AN ASSOCIATION (IF ANY) FROM ALL SELECTED OBJECTS

FIG. 17

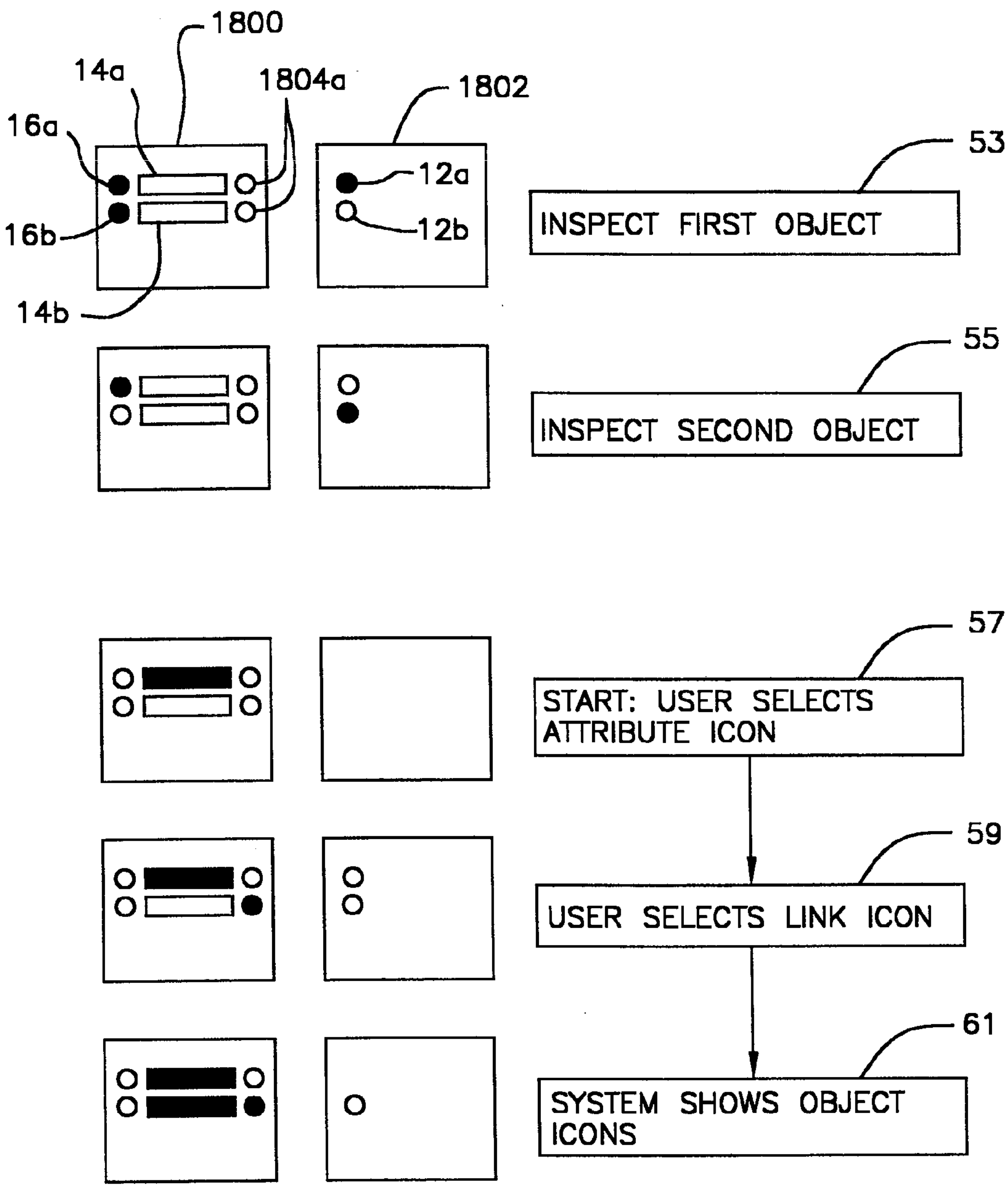


FIG. 18

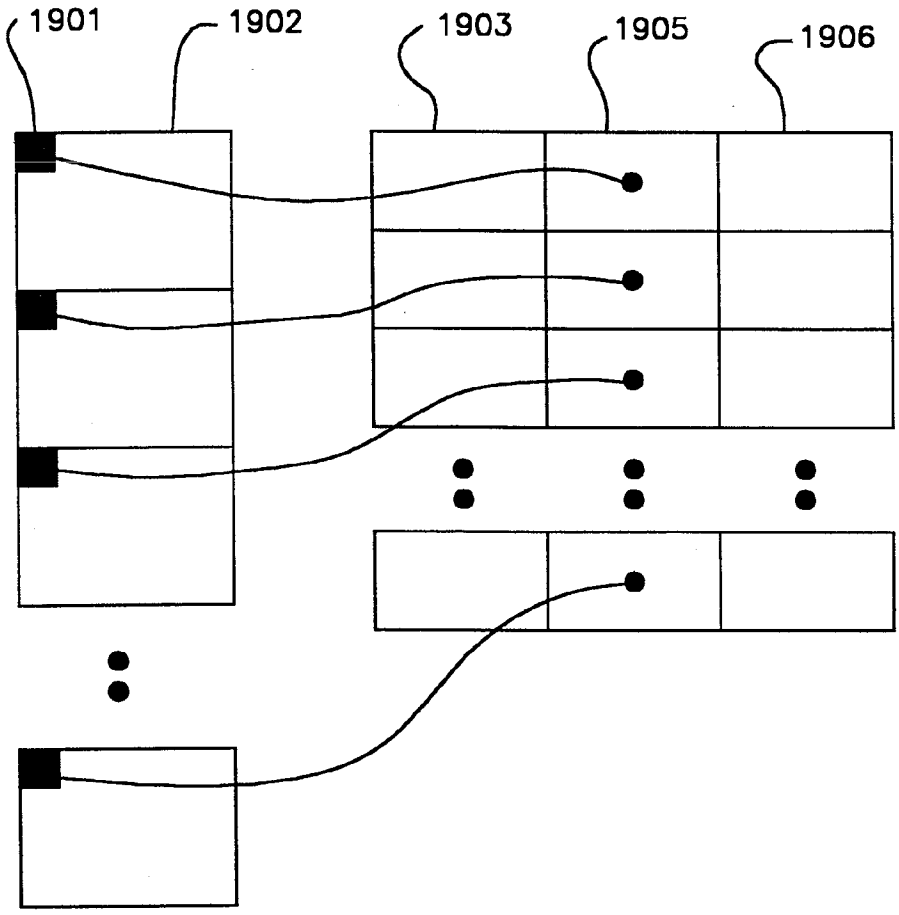


FIG. 19 (PRIOR ART)

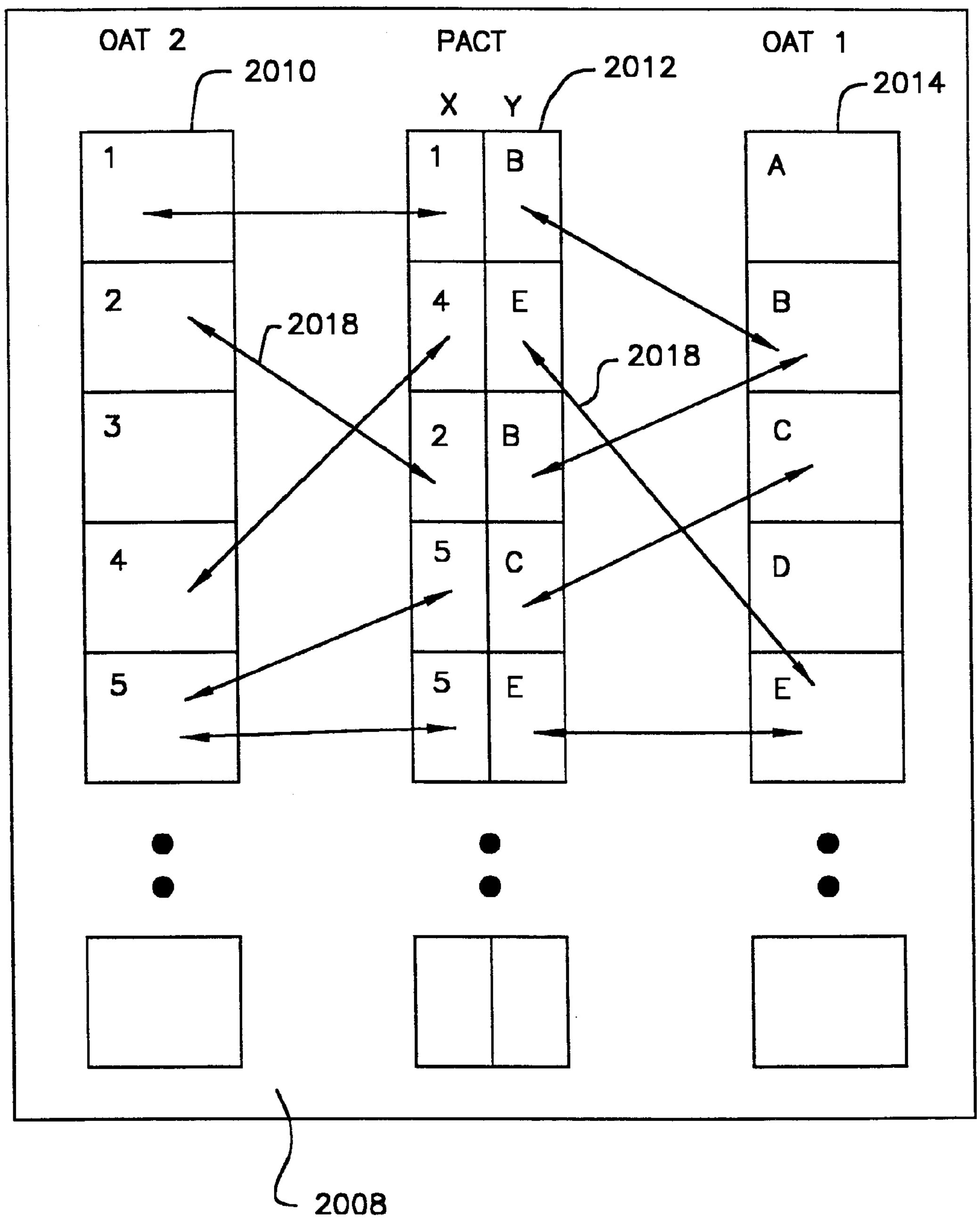

OAT 2
PACT
OAT 1
2010
2012
2014
X
Y
1
2
3
4
5
1
B
4
E
2
B
5
C
5
E
A
B
C
D
E
2018
2018
2008

FIG. 20

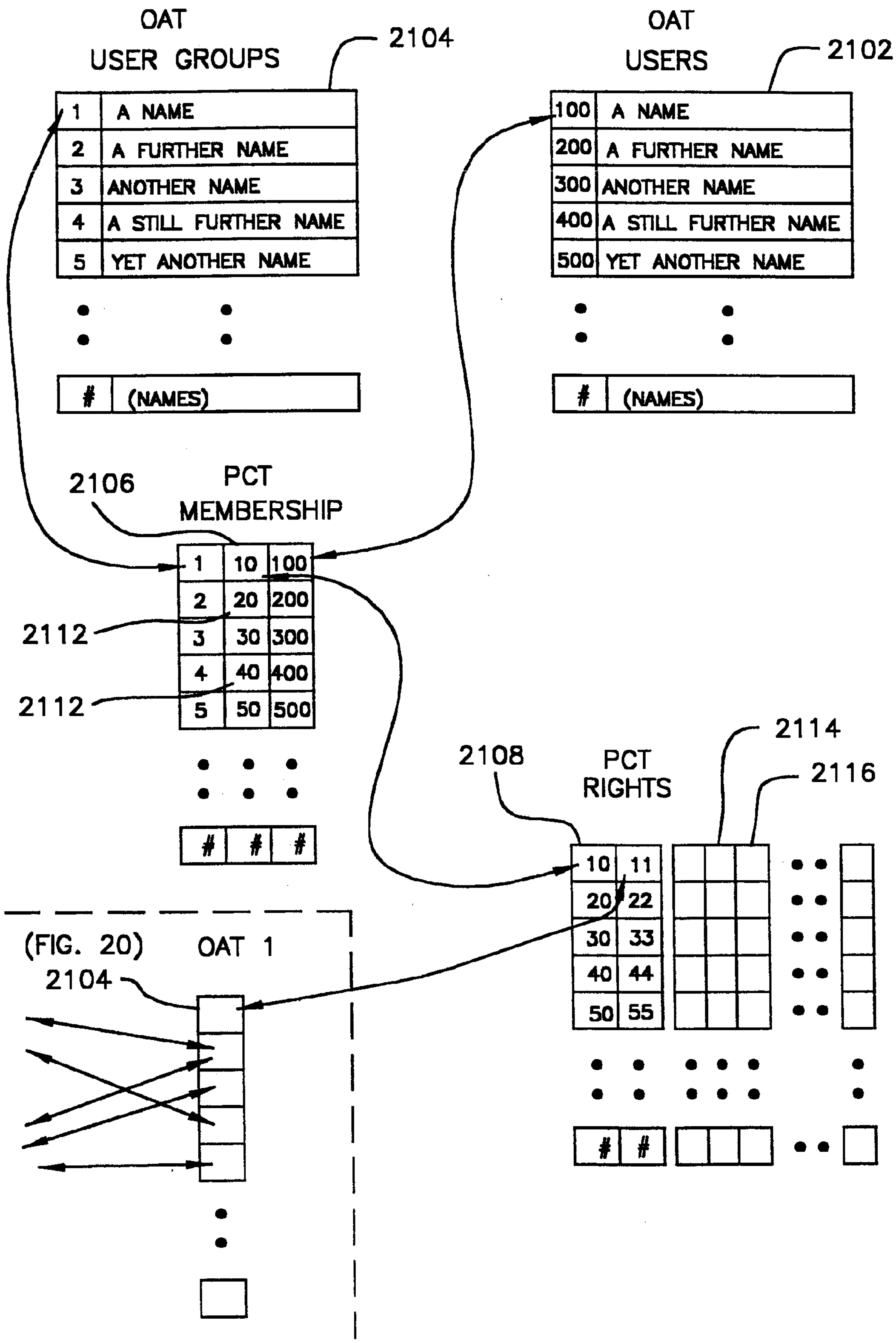

OAT
USER GROUPS
2104
1 A NAME
2 A FURTHER NAME
3 ANOTHER NAME
4 A STILL FURTHER NAME
5 YET ANOTHER NAME
(NAMES)
OAT
USERS
2102
100 A NAME
200 A FURTHER NAME
300 ANOTHER NAME
400 A STILL FURTHER NAME
500 YET ANOTHER NAME
(NAMES)
2106
PCT
MEMBERSHIP
1 10 100
2 20 200
3 30 300
4 40 400
5 50 500
2112
2112
# # #
2108
PCT
RIGHTS
2114
2116
10 11
20 22
30 33
40 44
50 55
# #
(FIG. 20)
OAT 1
2104

FIG. 21

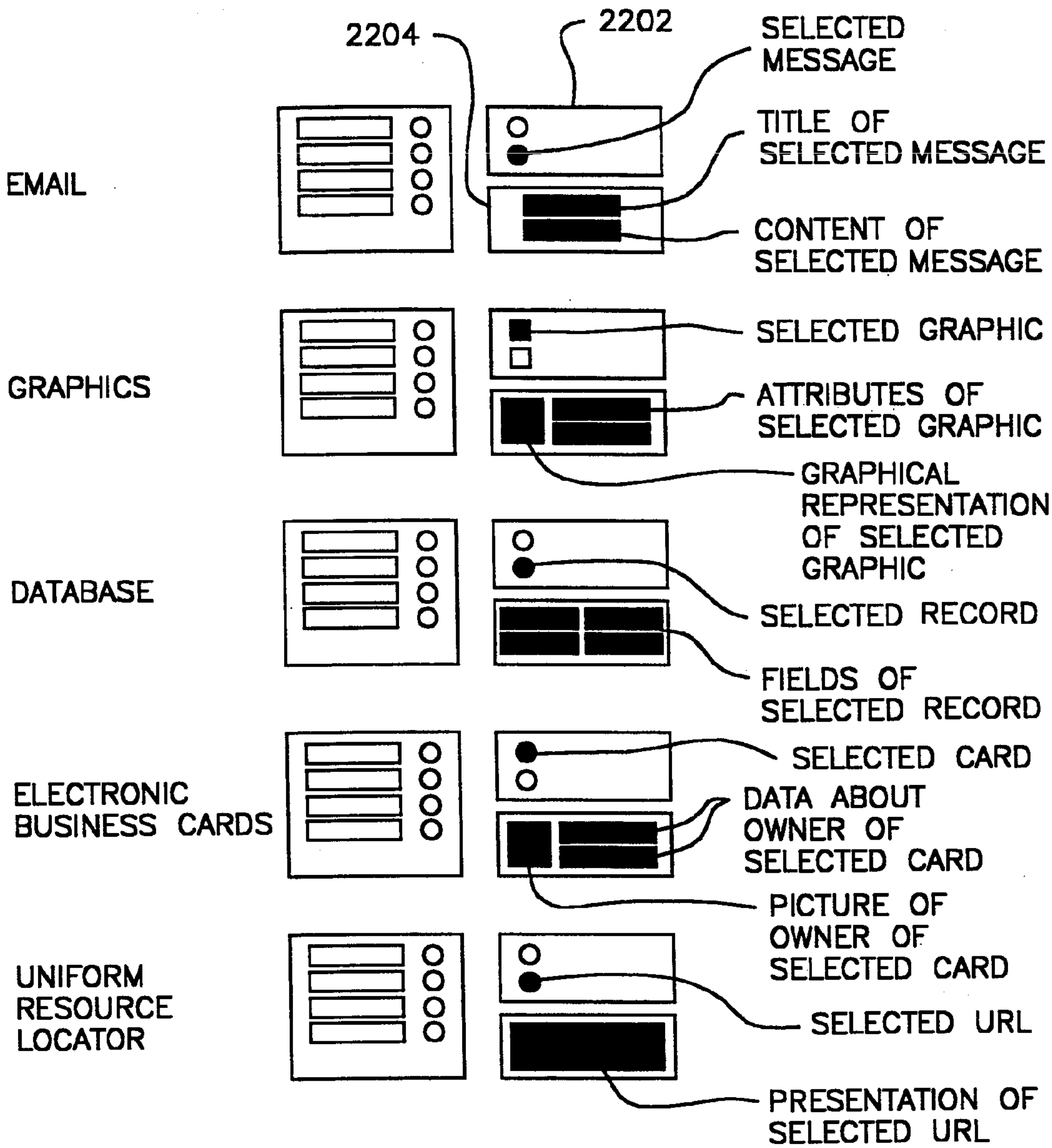

2204
2202
SELECTED MESSAGE
TITLE OF SELECTED MESSAGE
EMAIL
CONTENT OF SELECTED MESSAGE
SELECTED GRAPHIC
GRAPHICS
ATTRIBUTES OF SELECTED GRAPHIC
GRAPHICAL REPRESENTATION OF SELECTED GRAPHIC
DATABASE
SELECTED RECORD
FIELDS OF SELECTED RECORD
SELECTED CARD
ELECTRONIC BUSINESS CARDS
DATA ABOUT OWNER OF SELECTED CARD
PICTURE OF OWNER OF SELECTED CARD
UNIFORM RESOURCE LOCATOR
SELECTED URL
PRESENTATION OF SELECTED URL

FIG. 22